НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ ЗАХИСТУ ІНФОРМАЦІЇ

«На правах рукопису»
УДК _03.26.09:004.75

«До захисту допущено»
В.о. завідувача кафедрою
_____ М.М.Савчук
(підпис) (ініціали, прізвище)
“___” _____ 2020р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності_____Прикладная математика 113_____
(код і назва)

на тему: _____Аналіз уразливостей та побудова атак на протоколи криптовалют на основі технології <<графчейн>>_____
_____

Виконав (-ла): студент (-ка) __6__ курсу, групи _ФІ-83мн_____
(шифр групи)

_Полулях Володимир Романович_____ _____
(прізвище, ім’я, по батькові) (підпис)

Керівник_Ковальчук Людмила Василівна д.т.н,, проф_____ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____ _____ _____
(назва розділу) (науковий ступінь, вчене звання, , прізвище, ініціали) (підпис)

Рецензент_____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2020_року

Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»
Фізико-технічний інститут
Кафедра математичних методів захисту інформації

Рівень вищої освіти: другий (магістерський) за освітньо–професійною програмою

Спеціальність: 113 «Прикладна математика»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедрою
_____ М.М.Савчук
(підпис) (ініціали, прізвище)
«___»_____20_ р.

ЗАВДАННЯ
на магістерську дисертацію студенту

_____
(прізвище, ім'я, по батькові)

1. Тема дисертації Аналіз уразливостей та побудова атак на протоколи криптовалют на основі технології <<графчейн>>_____
_____
_____,
науковий керівник дисертації Ковальчук Людмила Василівна, д.т.н., проф.,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від 2020 р. № _____

2. Термін подання студентом дисертації _____

3. Об'єкт дослідження Процеси функціонування різних типів графчейнів
_____

4. Предмет дослідження (Вхідні дані – для магістерської дисертації за освітньо–професійною програмою)
Аналіз стійкості існуючих протоколів на основі графчейнів та побудова атак на ці протоколи . _____
_____

5. Перелік завдань, які потрібно розробити Провести аналіз властивостей запрованованих протоколів та визначити ступінь обгрунтованності їх стійкості. Провести аналіз недоліків технології блокчейн та можливі шляхи її удосконалення. Визначити найбільші вразливості запропонованих протоколів. На основі знайдених властивостей побудувати ефективні атаки на протоколи на основі графчейну. Уточнити оцінки параметрім функціонування протоколів _____
_____

6. Орієнтовний перелік ілюстративного матеріалу _____
_____
_____

7. Орієнтовний перелік публікацій _____
_____
_____

8. Консультанти розділів дисертації*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
	Обробити теоретичні матеріали стосовно технології блокчейн	15.12.2019	
	Обробити теоретичні матеріали стосовно технології графчейн та протоколів, заснованих на ній	08.03.2020	
	Провести аналіз протоколів на основі графчейну	20.04.2020	
	Оформлення результатів дослідження	01.05.2020	

Студент _____ Полулях В. Р.
(підпис) (ініціали, прізвище)

Науковий керівник дисертації _____ Ковальчук Л. В.
(підпис) (ініціали, прізвище)

* Консультантом не може бути зазначено наукового керівника магістерської дисертації.

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. Ігоря
СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

Звіт з виконання дослідження на тему

АНАЛІЗ УРАЗЛИВОСТЕЙ ТА ПОБУДОВА АТАК НА ПРОТОКОЛИ КРИПТОВАЛЮТ НА ОСНОВІ ТЕХНОЛОГІЇ «ГРАФЧЕЙН»

Виконав студент
групи ФІ-83мн
Полулях Володимир Романович

Науковий керівник:
д.т.н. Ковальчук Людмила Василівна

Київ — 2020

РЕФЕРАТ

Кваліфікаційна робота містить: 75 стор., 19 рисунків, 16 джерел.

Метою дослідження є уточнення та удосконалення методів криптоаналізу протоколів на основі технології графчейн. Об'єктом дослідження є процеси функціонування різних типів графчейнів. Предметом дослідження є аналіз стійкості існуючих протоколів на основі графчейнів, уточнення оцінок часу прийняття блоків та побудова атак на ці протоколи.

У цій роботі розглянуто протокол Біткойн та технологію блокчейн. Проаналізовано протоколи криптовалют на основі графчейнів.

Проведено аналіз проблем масштабування технології блокчейн. Знайдено недоліки в обгрунтуваннях стійкості протоколів на основі технології графчейн. На основі знайдених недоліків побудовано атаку подвійної витрати на один з протоколів. Отримано оцінки часу прийняття блоку для одного з протоколів.

БИТКОЙН, БЛОКЧЕЙН, ГРАФЧЕЙН, КРИПТОВАЛЮТИ

РЕФЕРАТ

Квалификационная работа содержит 56 с., 19 рисунков, 16 источников.

Целью исследования является уточнение и совершенствование методов криптоанализа протоколов на основе технологии графчейн. Объектом исследования являются процессы функционирования различных типов графчейнов. Предметом исследования является анализ устойчивости существующих протоколов на основе графчейнов, уточнения оценок времени принятия блоков и построение атак на эти протоколы.

В этой работе рассмотрены протокол Биткойн и технологию блокчейн. Проанализированы протоколы криптовалют на основе графчейнов.

Проведен анализ проблем масштабирования технологии блокчейн. Найдено недостатки в обоснованиях устойчивости протоколов на основе технологии графчейн. На основе найденных недостатков построено атаку двойной траты на один из протоколов. Получены оценки времени принятия блока для одного из протоколов.

БИТКОЙН, БЛОКЧЕЙН, ГРАФЧЕЙН, КРИПТОВАЛЮТЫ

ABSTRACT

The qualifying paper contains: 56 pages, 19 figures, 16 sources.

The purpose of the paper is to refine and improve the methods of cryptanalysis of protocols based on graphchain technology. The object of the research are the processes of graphchains different types functioning. The subject of the research is the analysis of the stability of existing protocols based on graphchains, refinement of block acceptance time estimates and the construction of attacks on these protocols.

In this paper, the Bitcoin protocol and blockchain technology are considered. Cryptocurrency protocols based on graphchains are analyzed.

The analysis of problems of scaling of blockchain technology was carried out. Deficiencies in the justifications for the stability of protocols based on graphchain technology were found. On the basis of the found shortcomings the double spend attack on one of protocols is constructed. Estimates of the block acceptance time for one of the protocols are obtained. BLOCKCHAIN, CRYPTOCURRENCY, GRAPHCHAIN

ЗМІСТ

ВСТУП

Актуальність дослідження. В умовах сьогодення блокчейн технології стають все більш популярними. Вони починають проникати у багато різних сфер нашого повсякденного життя таких, як фінанси та банківські перекази, міжнародні платежі, захист авторського права, електронне голосування, смарт-контракти, інтернет речей, анонімна передача повідомлень, кібербезпека, онлайн покупки, боротьба з DDoS-атаками, контроль якості продуктів харчування, регістрація лікарських засобів та багато іншого.

Проте, незаважаючи на велику кількість переваг над класичними технологіями, технології на основі блокчейну не позбавлені недоліків. Через ці недоліки блокчейн все ще залишається дещо екзотичним поняттям для пересічного громадянина. Однією із головних вад блокчейну є слабка здатність до масштабування. Збільшення розміру блоку чи підвищення швидкості додавання нових блоків суттєво знижує стійкість блокчейну до атак злоумисників. Це є основною причиною, чому такі криптовалюти, як Біткойн [3] не можуть в повній мірі замінити класичні системи електронно готівки як Visa [11] чи Mastercard [?].

Невдачі у спробах підвищення пропукної здатності криптовалют на основі блокчейну змусили дослідників розглядати нові способи децентралізованого зберігання транзакцій. Одний із найперспективніших, на нашу думку, є перехід від блокчейну, де блоки з транзакціями зберігаються у лінійнійному ланцюжку блоків, до узагальнень блокчейну, у яких транзакції чи блоки з транзакціями зберігаються у деревовидних структурах даних – графчейнах. На сьогодні відомо декілька протоколів заснованих на графчейнах: Blockchain-free, PHANTOM, GHOST, Tangle та ін. Розробники цих протоколів обіцяють більшу пропускну здатність в порівнянні з класичними криптовалютами. Проте стійкість протоколів на основі графчейну до атак злоумисників не досліджена належним чином.

Метою дослідження є уточнення та удосконалення методів криптоаналізу протоколів на основі технології графчейн. **Задачею дослідження** є аналіз стійкості протоколів на основі графчейнів до атак злоумисників, та побудова атак на такі протоколи.

Досягнення поставленої мети передбачає виконання таких **завдань дослідження**, які були виконані в роботі:

1) провести аналіз властивостей запрованованих протоколів та визначити ступінь обгрунтованності їх стійкості;

2) провести аналіз недоліків технології блокчейн та можливі шляхи її удосконалення;

3) визначити найбільші вразливості запропонованих протоколів;

4) на основі знайдених властивостей побудувати ефективні атаки на протоколи на основі графчейну;

5) уточнити оцінки параметрім функціонування протоколів.

Об'єктом дослідження є процеси функціонування різних типів графчейнів.

Предметом дослідження є аналіз стійкості існуючих протоколів на основі графчейнів та побудова атак на ці протоколи.

При розв'язанні поставлених завдань використовувались такі *методи дослідження*: методи дискретної математики, комбінаторики, математичної статистики, теорії імовірності, теорії складності алгоритмів.

Наукова новизна. У роботі отримані наступні результати:

1) показано, що всі протоколи мають недоліки в обгрунтуваннях стійкості;

2) побудовано ефективну атаку подвійної витрати на протокол Blockchain-free;

3) доведно твердження, які дозволяють виправити деякі помилки, допущені авторами зазначених протоколів.

Практичне значення. Результати даної роботи можуть бути використані при побудові нових та удосконалення вже протоколів на основі технології графчейн.

Апробація результатів та публікації. Основні положення дипломної роботи доповідалися і обговорювалися на XVIII Науково-практичній конференції студентів, аспірантів та молодих вчених «Теоретичні і прикладні проблеми фізики, математики та інформатики» (12–13 травня 2020 р., м. Київ).

1 ОПИС ТЕХНОЛОГІЙ БЛОКЧЕЙНУ ТА БЛОКГРАФУ

В даному розділі наводиться опис протоколу Біткойн [3], використання механізму знаходження консесусу Proof-of-Work при побудові криптовалют на блокчейні та проведено огляд основних недоліків даного підходу. Також розглянуто новий напрямок в технології блокчейн – графчейн, і його використання замість традиційного блокчейну.

1.1 Механізм знаходження консесусу proof-of-work

Доказ виконання роботи (англ. *proof-of-work*) це концепт, що був винайдений Dwork та Naor [1] для захисту мережевих систем від зловживання використанням послуг (спам, DDoS-атаки). Термін «proof of work» було вперше запропоновано та формалізовано Jakobson 1999 року [2].

Ідея PoW полягає у необхідності виконання на стороні клієнта деякої достатньої обчислювальної роботи (знаходження розв'язку задачі), результат якої легко перевіряється на строні сервера. Головною особоливістю є асиметрії витрат часу – вони значні для знаходження розв'язку та малі для перевірки його правильності.

Існує два типи proof-of-work протоколів.

1) *Задача-відповідь* протоколи припускають прямий інтеракативний зв'язок між клієнтом (запитувач) та сервером (постачальник). Постачальник визначає задачу, скажімо, знайти в множині елемент з деякою властивістю. Запитувач знаходить елемент, та відправляє його назад до постачальника. Оскільки задача обрана на стороні постачальника, складність може бути скорегована відповідно до його

навантаження. Обсяг роботи на клієнтській стороні може бути обмежена якщо протокол «задача-відповідь» має відомий розв'язок або якщо відомо, що він існує в обженому просторі пошуку.

2) *Розв'язок-верифікація* протоколи не припускають існування інтерактивного зв'язку. Таким чином задача має бути самовизначеною запитувачем і потім постачальник має перевірити і вибір задачі і вибір розв'язку.

Під час використанні у блокчейні (наприклад у Bitcoin [3]) Proof-of-work полягає у знаходженні майнерами значення криптографічної хеш-функції (SHA-256) від блоку з транзакціями, що відповідає заданим обмеженням, за допомогою перебору значень випадковго поля nonce. Складність майнінгу визначається спеціальним параметром (у Bitcoin – кількість нулів на початку гешу), що підбирається таким чином, щоб середній час майнінгу нового блоку був близьким до деякого визначеного параметру блокчейну (10 хвилин для Bitcoin). Для того, аби змінити в сформованому блоці необхідно виконати такий самий об'єм обчислень, як і при генерації нового блоку і всіх наступних. Однак перевірка цілісності ланцюжка вимагає одноразове обчислення гешу для кожного блоку, що є відносно дешевою операцією.

1.2 Протокол Біткойн

«Bitcoin» - це протокол, який реалізує незалежну валюту і платіжну систему [3]. Тобто, якщо розглядати інші електронні гроші або платіжні системи, наприклад, WebMoney, PayPal - це платіжні системи, які оперують уже існуючими валютами: долар, євро, фунт і т. Д. Bitcoin є і окремої валютою, і платіжною системою, яка керує цією валютою. Важливо розуміти, що це незалежна платіжна система, тобто немає організації, яка б контролювала її роботу.

Зауваження. Зауваження: далі ми будемо використовувати слова «Bitcoin» і «біткоіни», коли мова йде про платіжну мережі або протоколі, а слова «bitcoin» і «біткоіни», якщо мається на увазі валюта або відповідні монети.

1.2.1 Загальні Відомості

Тільки обмежене коло людей може достовірно знати, хто вперше запропонував дану технологію і назвав її Bitcoin. Офіційно відомо, що це був хтось під псевдонімом Сатоши Накамото. Можливо, цей хтось - одна людина, але є припущення, що за псевдонімом може стояти група людей. Сатоши зареєстрував домен bitcoin.org в 2008 році, випустив першу статтю, опублікував початкову версію вихідного коду протоколу. До 2011 року від псевдоніма Satoshi Nakamoto на відповідних форумах і в email-розсилках з'являлися повідомлення. Пізніше він написав, що вирішив зайнятися іншими справами і припинив публічні комунікації.

Проте варто визнати, що ідея створення Bitcoin [3] виникла у Сатоши Накамото зовсім не на рівному місці. У своїй статті «Bitcoin: A Peer-to-Peer Electronic Cash System» [3] він згадав два інших ключових проекти. Це були попередні спроби створити незалежну цифрову валюту: HashCash доктора Адама Бека (Adam Back) і B-Money інженера Вей Дая (Wei Dai). У першому з'явився підхід proof-of-work, спочатку створений для боротьби зі спамом в електронних листах, а в другому - модель мережі для розподіленого зберігання даних про транзакції і використання криптографічних підписів для відправки грошей.

Біткоіни - це перша успішна реалізація децентралізованої системи обліку. До нього були й інші спроби, наприклад, компанія DigiCash, яку заснував Д. Шаум в 1989 році. Вона була однією з найперших, які запропонували цифрові платежі зі спеціальною валютою, яка називалася Ecash. Користувачі платіжної системи були анонімні і банки не могли

відстежити їх рахунки. Проект підтримали тільки кілька інноваційних банків в США і один в Фінляндії, але оскільки було дуже складно переконати інші банки і Мерчант приймати анонімну валюту, проект згорнули.

Одним з перших кроків в розвитку Bitcoin [3] була, власне, публікація Bitcoin White Paper - 31 жовтня 2008 року. Перший вихідний код був опублікований в 2009 році, 3 січня, тоді ж був сформований і перший блок - в Bitcoin з'явилися перші 50 монет. Тому можна вважати цю дату запуском валюти. У лютому 2010 року відбувся перший обмін біткоіни на гроші.

Важливим в історії Bitcoin подією став форк Bcash (відгалуження від основної цифрової валюти), який відбувся 1 серпня 2017 року. Ще одне важливе оновлення – Segregated Witness – було активовано 24 серпня 2017 року. А 15 березня 2018 року побачив світ надбудову над Bitcoin для здійснення миттєвих платежів - Lighting Network.

На момент 2018 року по всьому світу біткоіни користуються мільйони людей, сотні тисяч компаній приймають його в оплату. На базі Bitcoin розробляється безліч проектів. Деякі країни, зокрема Японія, визнали його законним платежним засобом.

Використовуючи Bitcoin [3], можна відправити платіж кому завгодно і куди завгодно. Для цього потрібен доступ до мережі, цифровий гаманець і адреса одержувача. Велика кількість обмежень, характерні для звичайного міжнародного переказу, просто відсутні. Ніяких додаткових дозволів для здійснення платежу не потрібно. Тому Bitcoin [3] зацікавив людей, які підтримували анархізм, ліберальні ідеї, повну приватність і т. п. Саме в цих колах Bitcoin [3] здобули популярність спочатку. Пізніше до них приєдналися комп'ютерні гіки (шіфропанкі, хакери), а також вчені, для яких Bitcoin був об'єктом дослідження, який дозволяв проводити цікаві експерименти. Як тільки ціна біткоіни стала хоч якось помітною, він почав привертати увагу підприємців і спекулянтів, які намагалися заробити на ньому. Для звичайної людини

інтерес представляє відсутність необхідності реєструватися і можливість здійснювати платежі без залучення третіх сторін.

Перш ніж продовжити огляд протоколу Біткойн розглянемо деякі необхідні для поняття.

Визначення 1.1. Bitcoin-адреса – велике унікальне число, яке служить для ідентифікації отримувача монет.

Визначення 1.2. Bitcoin-транзакція – цифрове заява, щодо зміну власника монет.

Визначення 1.3. Bitcoin-протокол – набір правил, за якими взаємодіють вузли мережі Bitcoin [3].

Визначення 1.4. Bitcoin-мережа – peer-to-peer мережу, що складається з тимчасових вузлів.

Визначення 1.5. Вузол мережі – це комп'ютер із запущеним ПЗ для роботи з іншими комп'ютерами за певномим протоколом, також їх називають повними вузлами мережі (full nodes).

У найпростішому випадку адреса прив'язаний до однієї пари ключів (відкритий ключ і особистий ключ), яка використовується для формування та перевірки електронного підпису. Особистий ключ використовується для запевнення транзакцій. Важливо відзначити, що простір всіх можливих адрес величезне, а простір можливих особистих ключів ще більше. При коректної генерації, вгадати або підібрати особистий ключ до чужого адресою практично неможливо. Важливо не плутати поняття адреси з поняттям аккаунта, оскільки в рамках протоколу Bitcoin акаунтів не існує.

Також важліво розуміті яка інформація зберігається в транзакціях. Будь-яка транзакція містить дані про походження монет, які витрачаються (посилання на транзакції, де ці монети були отримані), докази володіння монетами, адреси нових власників і суми перекладів.

1.2.2 Комісія та консенсус Біткойн

Модель транзакцій в Bitcoin [3] передбачає комісійні збори, які оплачуються біткоіни. Комісію включає сам відправник в момент створення транзакції, і за замовчуванням вона повинна бути вище певного порогового значення. Хоча на практиці користувач може встановити її рівною нулю і така транзакція теоретично буде вважатися правильною. Цю комісію в якості додаткової винагороди отримують ті учасники, які підтверджують транзакції.

Процес підтвердження транзакцій передбачає те, що учасники попередньо перевіряють їх, після чого погоджують, які транзакції будуть вважатися правильними. Для підтвердження транзакція повинна отримати згоду більшості учасників. У процесі підтвердження транзакцій в Bitcoin [3] може взяти участь будь-який бажаючий. Набір правил за Яким відбувається Додавання Нових блоків називається emph консенсусом.

Визначення 1.6. *Блокчейн* (англ. *blockchain*) – вибудований за певними правилами неперевного зростаючого ланцюжка блоків, що містять деяку інформацію. Зв'язок між блоками реалізовано не тільки за допомогою нумерації, а тим, що кожен блок містить власний геш та геш попереднього блоку. Таким чином блокчейн є зв'язним списком.

Цілісність та непідробність блокчейну гарантується його будовою, адже для підробки одного його блоку необхідно змінювати й усі наступні блоки. Також на користь надійності блокчейну виступає той факт, що його копії зберігаються на багатьох незалежних вузлах мережі.

Для функціонування блокчейні зазвичай використовують peer-to-peer мережі, що утворюються з незалежних, рівноправних вузлів (майнерів).

Цікава особливість Bitcoin полягає в тому, що існує можливість в будь-який транзакції показати, звідки беруться монети, тобто

виконується посилання на попередню транзакцію, із зазначенням її хеш-значення. Таким чином, відбувається перевірка історії походження монет, які передаються.

Для того аби транзакція була додана в блок для майнингу, проводиться її верифікація. Розглянемо основні етапи перевірки транзакції:

1) перевірка існування монет в системі

2) перевірка на подвійну витрату

3) перевірка доказів володіння монетам

Працює це таким чином. Щоб витратити монети, ви повинні показати, де ви їх отримали, і довести, що саме ви ними володієте.

Якщо походження монет не викликає додаткових питань, відсутня інша транзакція, яка витрачає ці монети, і ви дійсно довели, що вони ваші, то залишається тільки дочекатися підтвердження цієї транзакції іншими учасниками мережі.

Настав час заглибитися в процес створення блоку. Рішення проблеми з підробленими блоками полягає в наступному: блок вважається правильним, якщо на його створення було витрачено задану кількість обчислювальних ресурсів. Це означає, що користувач повинен надати рішення ресурсномісткої завдання, щоб всі інші учасники могли перевірити і прийняти його блок.

Наразі зловмисник вже не може відволікати інших учасників великою кількістю підроблених блоків. Попутно з цим вирішуються питання частоти появи нових блоків і черговості валідаторів в формуванні наступного блоку. Працює це таким чином. Починають формувати новий блок все, але тільки той, хто вирішив ресурсномістких завдання першим, отримує право запропонувати свій блок іншим.

Очевидно, що чим більше у користувача обчислювальних потужностей, тим частіше він стає першим серед інших бажаючих. Якщо конкретніше, то ймовірність стати першим дорівнює відсотку обчислювальних ресурсів учасника від всіх ресурсів, задіяних в мережі.

Процес створення нових блоків:

1) необхідно надати розв'язок ресурсоємкої задачі;

2) той, хто вирішив задачу першим, розсилає решті свій блок;

3) Імовірність стати першим залежить від частки ресурсів учасника у всіх обчислювальних ресурсах, задіяних в мережі, а також від затримок в каналах передачі даних

По суті розв'язання ресурсноємної задачі - це і є майнінг. Необхідно розуміти, що це завдання має однакову складність для всіх вузлів мережі. Майнінг дуже важливий для Bitcoin і чесні користувачі займаються цим з метою підтримки надійності процесу підтвердження транзакцій. Справедливо, що той, хто контролює більше потужності створює блоки частіше. Але і в цій ситуації зловмисник вже не може нав'язувати свою альтернативну думку, не маючи більшої частини обчислювальної потужності всієї мережі.

Але що відбувається коли 2 майнери пропагують блок зі спільними транзакціями? Якщо один учасник не згоден з блоком іншого, то абсолютно нормальним вважається створити альтернативний блок на тій же висоті ланцюжка блоків. Таким чином, в загальному вигляді розв'язок конфліктів між учасниками має наступний вигляд:

1) незгідний учасник формує альтернативний блок;

2) альтернативні блоки можуть включати однакові транзакції;

3) вузли мережі зберігають обидва варіанти;

4) решта учасників формують блоки, продовжуючи одну з версій ланцюжка;

5) перемагає ланцюжок з найбільшою довжиною (найбільшою кількістю роботи, вкладеному в його побудову);

Таким чином, факт незгоди одного з учасників може спричинити за собою ситуацію, коли буде сформовано два блоки, які посилаються на один попередній. Такі блоки можуть включати навіть однакові або конфліктуючі транзакції. При цьому вузли мережі можуть зберегти обидва запропоновані варіанти, якщо вважатимуть їх правильними, але

кожен з них повинен для себе визначити, на базі якого з альтернативних блоків створювати наступний. Таким чином, інші учасники роблять свій вибір, формуючи блоки на одному з існуючих блоків, вказуючи посилання на нього в своєму новому блоці для продовження конкретної версії ланцюжка. А правила протоколу вказують на те, що з двох правильних версій пріоритет потрібно віддавати тій, на створення якої було витрачено більше обчислювальних ресурсів.

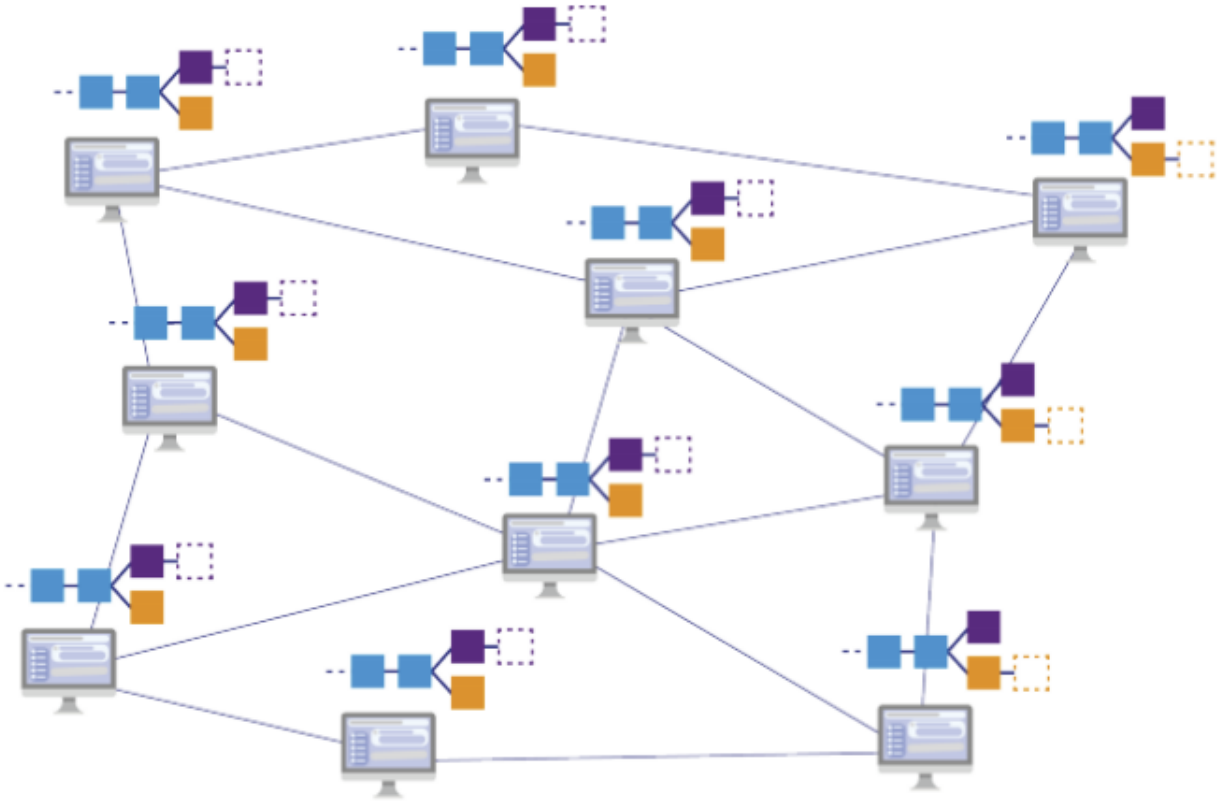

Рисунок 1.1 – Виникнення конфлікту між згенерованими блоками

Процес розв'язання суперечностей можна зобразити схематично. Це вирішується просто. Існує мережа вузлів, в локальних копіях бази даних яких існує два альтернативних блоку на одній висоті. Умовно позначимо, що користувачі зліва вирішили вибрати верхній блок в якості основного, а користувачі справа - нижній (рис. **??**). Таким чином, всі продовжують працювати над створенням наступного блоку, підтримуючи різні версії.

В наступний момент часу, якийсь користувач створив і запропонував

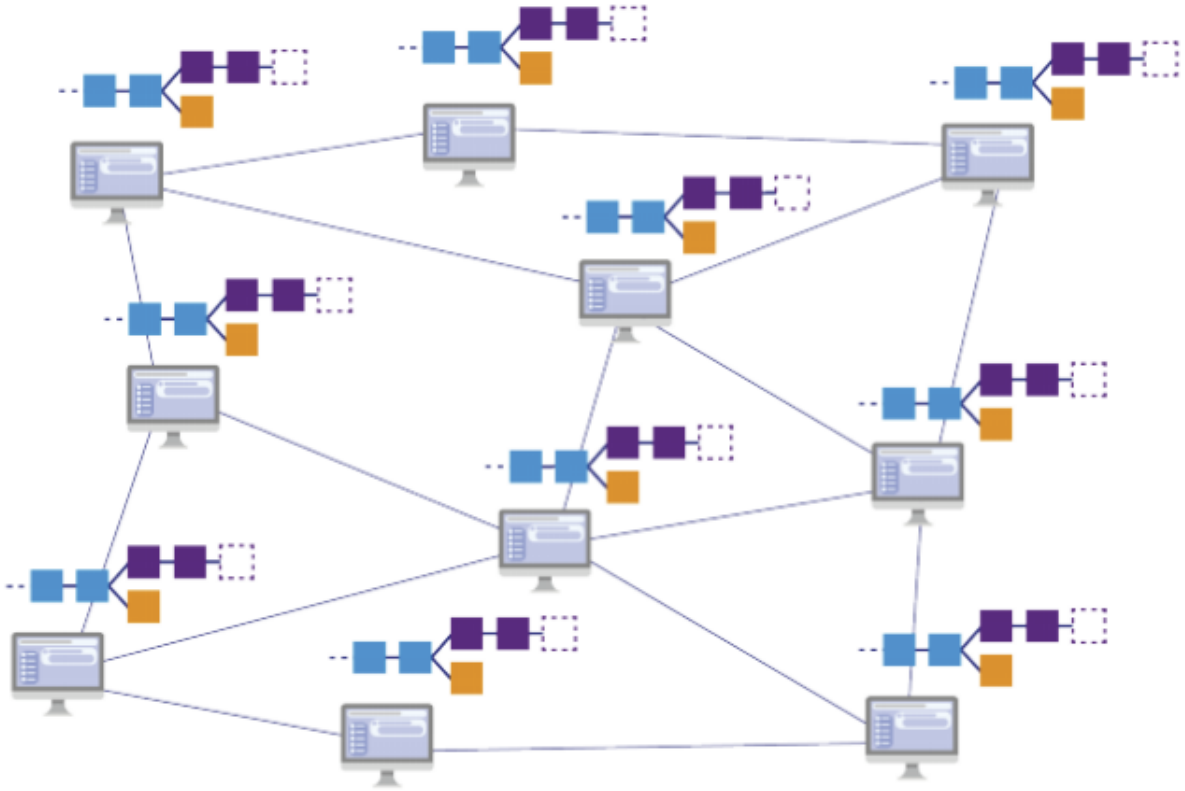

Рисунок 1.2 – Розв'язання конфлікту між згенерованими блоками

новий блок, який посилається на верхній з двох альтернативних. При цьому його пропозиція була прийнята всіма іншими учасниками мережі. І навіть ті учасники, які спочатку вибрали інший альтернативний блок, перевірили і прийняли ланцюжок, яка довший (рис. 1.2).

Це так зване правило найдовшого ланцюжка. У ньому йдеться, що учасник повинен вибирати найдовшу ланцюжок з усіх, які він вважає правильними, і вважати її основною. А якщо більш строго, то вибирається той ланцюжок, для створення якої було виконано більше роботи.

До того ж, важливо розуміти, що чесний учасник переключиться на довжелезну ланцюжок тільки в тому випадку, якщо вона була побудована за правилами протоколу. Це не дозволяє зловмисникам порушити початкові правила Bitcoin, навіть якщо вони будуть мати велику обчислювальну потужність.

Еще один важный вопрос – это мотивация пользователей решать

ресурсоемкие задачи, создавать новые блоки и подтверждать транзакции, не оставляя шансов злоумышленникам.

Ми вже говорили про процеси формування блоків, емісії монет і про комісії за підтвердження транзакцій. У Bitcoin ці процеси дуже тісно пов'язані один з одним. Справа в тому, що за правилами протоколу творець блоку може відправити на свою адресу певну кількість монет, взявши їх з нізвідки. Це і є емісійні монети. Підсумкова сума винагороди розраховується як емісійні монети плюс сума комісій всіх транзакцій, включених в цей блок.

Таким чином, в 2018 році винагорода за створення одного блоку становить 12.5 монет плюс комісійні збори. Відзначимо, що з міркувань безпеки валідатор не отримує при цьому винагороду відразу після створення блоку. Існує спеціальний параметр - coinbase maturity, який вказує на мінімальну кількість підтверджень транзакції, в якій валідатор отримує винагороду. У Bitcoin цей параметр має значення 100, отже, після створення блоку необхідно дочекатися появи ще 99 блоків, які будуть підтверджувати цей, перш ніж винагороду стане доступним.

1.3 Потенційні вразливості та атаки на блокчейни, засновані на PoW

Розглянемо основні типи атак на блокчейни.

Атака 51% На початку свого існування більшість криптовалют мають вразливість до так званої «атаки 51%». Якщо в зловмисник володіє біль ніж половиною усіх обчислювальних потужностей, то у нього з'являється можливість підтверджувати тільки власні блоки, нехтуючи чужими. Таким чином він має можливість отримувати 100% усіх нових койнів та блокувати будь-які транзакції. Одним з можливих варіантів атаки є перепис усього ланцюжка блоків починаючи с деякого моменту у

минулому. Як результат, він може зробити власний форк та перегнати основний ланцюжок блокчейну зробивши його невалідним.

Подвійні витрати *Подвійна витрата* – повторна передача одних і тих самих активів. Існує декілька різновидів даної атаки, розглянемо один із них. Зловмисник виконує транзакцію X проводячи тим самим оплату товару. Одночасно він створює альтернативний блок з транзакцією Y, якою переводить ті ж монети, що й у транзакції X, але переводить їх на свій рахунок. Потім зловмисник починає майнинг блоків прихованого ланцюжка блоків над блоком з транзакцією Y. Після отримання товару або послуги злоумисником від продавця, у найбільш вигідний для себе момент, зловмисник підміняє основний ланцюжок в якому міститься транзакція X, своїм ланцюжком з транзакцією Y.

Атака розділення На відміну від звичайної атаки подвійної витрати, у випадку атаки розділення зловмисник не створює форк, таємно очікуючи моменту опублікування з метою отримання найбільшої вигоди для себе. Він діє відкрито і ділить загальнодоступний журнал транзакції, на початку форку, таким чином включаючи інших майнерів розділитися між двух ланцюгів. Більше того, в такому випадку не тільки зловмисник вносить блоки в форкнуті ланцюги, але й чесні майнери.

На сьогодні головним принципом, за яким відбувається функціонування криптовалют є внесення змін до загальнодоступного журналу транзакцій (transaction ledger) шляхом голосування вузлів мережі за правилами, що описуються механізмом консенсусу. Власне журнал транзакцій формується при децентралізованому виконанні роботи учасниками мережі. До консенсусу висувається вимога стійкості до атак злоумисників, чиї обчислювальні ресурси є обмеженими.

Найбільш відомим та тим, що має найбільшу капіталізацію є протокол Bitcoin [3], що базується на механізмі консесусу proof-of-work. Проте він має суттєвий недолік, що полягає у надзвичайно низькій пропускній здатності, а саме 500 000 транзакцій за день, що не дає можливість використовувати [3], як повноцінну платіжну систему. В

порівнянні найпопулярніша класична платіжна система Visa [11] виконує порядку 150 000 000 транзакцій на день.

Така різниця у порівнянні криптовалют з класичними системами електронної готівки є обумовлена обмеженнями, що накладаються на на швидкість генерації нових блоків та на розмір блоку. Намагання підвищити середню кількість траназакцій за одиницю часу за допомоги варіювання цих двох параметрів призводить до збільшення вразливості блокчейну до атак злоумисників або частокової втрати децентралізованності криптовалюти. Так укрупнення блоків веде до збільшення часу їх затримки, що веде до збільшення імовірності успіху атак подвійної витрати та розгалудження. Підвищення швидкості генерації нових блоків призводить до збільшення імовірності ненавмисного форку, а також збільшення імовірності успіху атаки добре синхронізвованого злоумисника.

1.4 Основні проблеми масштабування блокчейну

На сьогодні ймовірність ненавмисного форку в реальних криптовалютах дуже низька завдяки правилу найдовшого ланцюжка, описаного в протоколі Накамото [3]. Проте даний спосіб знаходження консенсусу, що вважається за класичний дослідниками криптовалют та блокчейнів, є надзвичайно чутливими до збільшення розміру блоку або підвищення швидкості генерації нових блоків. Таким чином, наслідком зміни названих параметрів є зміна реального балансу обчислювальної потужності між частками чесних майнерів та злоумисників на користь останніх. Результатом даного розбалансування є створення ситуацій, коли достатньою умовою виникнення атаки с імовірністю успіху рівною 1, є наявність у злоумисника значно менших обчислювальних потужностей ніж 50% від загального об'єму обчислювальних потужностей. Таким

чином, мережа, що працює за протоколом найдовшого ланцюжка з високою швидкістю генерації блоків та великими блоками ймовірна успішна подвійна трата при наявності у злоумисника 25% обчислювальних потужностей. В даних умовах очевидною є актульність пошуку шляхів удосконалення механізму консенсусу зі збереження стійкості до атак злоумисників та збільшенням швидкості обробки нових тразакцій. Досягнення одночасного виконання цих умов слугуватиме унікальним торговою пропозицією криптовалюти на ринку, та дозволить їй стати монополістом. Саме тому сьогодні існує неприхований інтерес серед дослідників стосовно масштабування блокчейну, що не шкодить його привабливим для інвесторів властивостям.

Для забезпечення стійкості до атаки подвійної витрати, у блоках має міститися інформація про виконану роботу. Складність цього завдання налаштована адаптивно, так що блок створюється приблизно раз на 10 хвилин у всій мережі. Після створення блоки розповсюджуються через мережу. 10-хвилинний інтервал дозволяє блокам (як правило) поширюватися на переважну більшість вузлів до моменту створення іншого блоку. Якщо вузол отримує два суперечливі блоки, які були створені віддаленими вузлами, не знаючи про роботу один одного (або, можливо, зловмисника), він вирішує конфлікт шляхом вибору блоку, що відноситься до найдовшого ланцюга блоків, та прийняття його. Попередній аналіз протоколу Сатоші Накамото [12] показує, що поки зловмисник утримує менше за 50% обчислювальної потужності в мережі, імовірність того, що атака подвійної витрати буде успішною, з часом експоненційно зменшується. Це дозволяє вважати платежі прийнятим та незворотним через певний період часу. Однак аналіз передбачає, що блоки надсилаються через мережу набагато швидше, ніж вони створюються, і тому він неправильно підходить до сценарію, в якому мережею обробляються багато транзакцій (що вимагає частого створення більших блоків).

Нинішня відносно невелика кількість транзакцій у Біткойн в

основному пов'язана з невеликою кількістю користувачів. Після того, як кількість користувачів збільшиться, системі потрібно буде масштабувати обробку транзакцій з більш високою швидкістю, і попередні гарантії безпеки можуть перестати бути дійсними. Ми досліджуємо, наскільки протокол сприйнятливий до атаки подвійної витрати, коли в секунду оброблятиметься більше транзакцій. Зауважимо, що більші розміри блоків або частіші події створення блоків (які потрібні для збільшення пропускної спроможності транзакцій) призводять до більшої кількості конфліктів між блоками, що сильно знижує рівень захищенності до атак.

З метою зменшення негативних ефектів від зростання швидкості обробки транзакцій, членами Біткойн спільноти було запропоновано декілька методів компресії блоків, таких як передача хешей транзакцій у блоках (майже 16-кратне зменшення розміру) або застосування таблиць пошуку Блума для передачі відмінностей між підмножинами транзакцій про які відомо вузлам [?]. Інший підхід полягає у використанні ненадійних каналів транзакцій поза ланцюга, які повільно відпускають гроші іншим учасникам за рахунок хвилинних оновлень шляхом оновлення транзакції, яка здійснюється лише в блок-ланцюзі, тільки коли буде передана розумна сума грошей [?]. Такий підхід має деякі недоліки: гроші повинні бути заблоковані протягом часу існування каналу, він дозволяє лише сукупність транзакцій між двома сторонами, які підтримують канал, і, нарешті, він не завжди корисний для інших протоколів, побудованих на блокчейнах (таких як Ethereum), де окремі оновлення не можуть бути агреговані аналогічно.

Нами проведено аналіз основних проблем, з якими стикаються дослідники при розробці нових протоколів, що ставлять собі на меті підвищення пропускної здатності розподіленої мережі, стикаться з настпуними проблемами:

1) Повільна обробка транзакцій. Є два напрями вирішення даної проблеми: підвищення швидкості генерації нових блоків та збільшеннння їх розміру.

2) Тривалий час підтвердження транзакції. Протокол Bitcoin [3], для підтведження транзакції і ,відповідно, для додавання блоку, що її містить в основний ланцюжок вимагає, щоб над цим блоком було побудовано 6 «блоків підтвердження». Середня швидкість генерації одного блоку в даній мережі становить 10 хвилин, таким чином час підтвердження транзакції є випадковою величиною з середнім значенням в 1 годину. Проте часто трапляється, що час підтвердження транзакції досягає значно більших значень через високу дисперсію.

3) Відносна децентралізація блокчейну у реальних умовах. На практиці, через деякий час існування блокчейну, ймовірність успіху окремого майнеру низька через те, що його ресурси майнінгу становлять малу частку загальних ресурсів. Така ситуація змушує майнерів об'єднуватися у величезні майнінгові пули. Необхідні для майнінгу ресурси – це депозит у криптовалюті (при механізмі PoS), або обчислювальні потужності (для PoW), або інший необхідний ресурс для іншого механізму консенсусу.

4) Збереження лінійного порядку блоків. Дана властивість здається неочевидною при огляді необхідних властивостей протоколів криптовалют, проте має вирішальне значення для забезпечення реалізації механізмів функціонування смарт-контрактів у передбачених для цього протоколах. Класичному блокчейни володіють даною властивістю за означенням – інша ситуація з блокграфами. На практиці дослідники докладають багато зусиль для забезпечення лінійності.

5) Збереження рівня стійкості протоколів до основних типів атак на блокчейни. Через новизну та загальність такого поняття як графчейн, а також через велику кількість можливих варіацій реальних протоколів, не існує способів доводення стійкості протоколів до атак злоумисників. Для блокчейнів методи та відповідні результати, що доводять їх стійкість до атаки подвійної трати, splitting-атаки, атаки відмови додавання конкретного блоку та атаки на зміну лінійного порядку блоків. Знання про стійкість графчейнів не відрізняються комплексністю і представлять

окремі оцінки стійкості при загальних початкових припущеннях: для атаки подвійної трати отримано аналітичні оцінки верхньої межі імовірності успіху при найзагальніших припущеннях, для інших атак існують асимптотичні оцінки з неперервним часом.

Таким чином, наведені вище проблеми не є тривіальними та вимагають значних зусиль при проектуванні та доведенні стійкості й ефективності нових протоколів.

Для подолання цих труднощів дослідники відходять від класичного блокчейну і переходять до протоколів на основі деревовидних структур. На нашу думку найточнішими термінам, що відображають конструкцію, якою автори намагаються узагальнити блокчейн є «графчейн» та «блокграф». Дослідники використовували пропонували різні назви: ґешграф, графчейн, блокграф, DAG (directed asyclic graph) та інші. Найвживанішою є аббревіатура DAG – направленний ациклічний граф. Ми теж будемо вживати цей термін як синнонім до блокграфу. Нами були розглянуті наступні протоколи на основі блокграфів: GHOST [10], Blockchain-free[5], PHANTOM [9], Tangle [7].

Висновки до розділу 1

У даному розділі наведено історії розвитку протоколу Біткойн [3]. Подано перелік його основних понять та подано принципів функціонування криптовалюти. Розглянуто принципи роботи технології блокчейн, як узагальненя протоколу Біткойн [3]. Наведено основні недоліки протоколів побудованих на блокчейні. Наведено визначення графчену – узагальнення технології блокчейн. Зроблено висновок про нетривільність задачі побудови протоколу консенсусу на блокграфах, що має усунути недоліки протоколів криптовалют на блокчейнах зі збереженням їх гарних властивостей.

2 АНАЛІЗ ПРОТОКОЛІВ, ЗАСНОВАНИХ НА ТЕХНОЛОГІЇ ГРАФЧЕЙН

Другий розділ присвячено аналізу протоколів, заснованих на технології графчейн (інша назва блокграф). У цій частині наведено основні позначення, проаналізовано правила побудови блокграфів майнингу та винагороди майнерів, а також правила вирішення конфліктів наступних протоколів: GHOST, Blockchain-free, PHANTOM, TANGLE. Основною ідеєю, яку мали на меті автори протоколів є підвищення пропускної спроможності протоколів. Разом із тим, висвітлено деякі недоліки запропонованих протоколів.

2.1 Blockchain-free Cryptocurrencies

Головною іновацією, запропанованою у протоколі Blockchain-free [5], є підхід, за яким результуючий граф формується не з блоків, а з транзакцій. Таким чином, саме на транзакції покладено функцію підтвердження попередніх транзакцій. Ще одна відмінність від протоколів на основі блокчейну полягає у тому, що підтверджені транзакції зберігаються не в ланцюжку блоків з транзакціями, а у опорному графі (lean graph) з самих лише транзакцій – графчейн.

2.1.1 Правила функціонування графчейну

Крім власне «криптовалютної» інформації, кожна транзакція x містить вказівники на двох своїх «батьків» – валідних транзакцій з правильним PoW, що мають спільні з x ребра у графчейні. Якщо

майнери діють раціонально, то графчейн має вигляд передбаченого протоколом опорного графу.

Транзацкії виконують 2 функції: власне *транзакційну* та *структурну*. Відповідно кожна транзакція складається з:

1) *транзаційного* компоненту, що відображає корисне для користувача навантаження – *пейлоад* – готівка, смарт-контракти, тощо. Фреймворк майже не спеціалізує та не накладає обмеження на даний компонент, окрім можливості отримання винагороди майнерам.

2) *верифікаційний* компоненту (метадані), що відповідає за системний бік транзакції. Фреймворк покладається на цей компонент, для того аби побудувати графчейн, використовуючи proof-of-work, предків транзацкій, тощо.

У кожної транзакції є певна «винагорода», яка передбачається майнерам за виконання PoW. Значення винагороди є пропорційним до розміру транзакції. Обробляти «свіжі» транзакції вигідніше за «старі». Транзакція бере винагороду не тільки у «батьків», але й в інших «предків», у яких винагороду вичерпано не повністю. Розмір винагороди, яку транзакція бере у своїх предків, залежить від PoW даної транзакції і PoW предків. Забороняється чіплятися до транзакцій у яких винагороду вже вичерпано, однак є можливість чіплятися не тільки до «листків» графчейну, головне – ненульова винагорода.

Винагорода збирається з найстарших предків двох батьків шляхом обходу графу по всім транзакціям, у яких винагороду ще не вичерпано. На кожній ітерації з предка знімається вся можлива винагорода, що йде на покриття винагороди, призначеної за дану транзакцію, яка базується на об'ємі роботи необхідного для виконання PoW.

Верифікація направлена на більш нові транзакції. Таким чином, прямі валідації старших транзакцій будуть заборонені, як тільки їх винагороду буде вечерпано. Транзакції з більшою винагородою будуть приваблювати великих майнерів – для швидкого прийняття транзацкії до графчейна. Транзакції з ненульовою, але низькою винагородою будуть

включені до графчейну меньшими майнерами, ймовірність змагання між якими поступово зменшується з часом через зменшення винагороди.

Неправильні транзакції (подвійна трата, некоректне форматування, несвоєчасні виплати тощо) мають бути відкинуті більшістю.

2.1.2 Основні позначення

Визначення 2.1. Наведемо визначення транзакції з роботи [5]: Нехай $Payments$ – інформація про платіж, x_l, x_r – попередні транзакції, c – складність, f – комісія, m – монета, s – розв'язок PoW. Тоді транзакцією x_i назвемо

$$x_i = [Payments_i, x_l, x_r, c_i, f_i, m_i, s_i]. \tag{2.1}$$

Транзакції реалізують усі ролі в протоколі: передають електронну готівку, перерозподіляють винагороду, додають комісію і ,головне, – підтверждують правильність попередніх транзакцій.

Визначення 2.2. Наведемо визначення поняття $T - POSET$ з роботи [5]: нехай P – множина транзакцій. Для деяких різних $t, t' \in P$ запишемо $t \prec t'$ тоді і тільки тоді, коли t знахотиться в схемі PoW t' (запис $t \in past(t')$). Таким чином отримуємо частко вопрядковану множину транзакцій – Transaction Partially Ordered Set ($T - POSET$).

Визначення 2.3. Нехай $x \in P$, $P - -T - POSET$, тоді минулим x будемо називати множину

$$past(x) = y_i : y_i \prec x.$$

Визначення 2.4. Нехай $x \in P$, $P - -T - POSET$, тоді майбутнім x будемо називати множину

$$future(x) = y_i : y_i \succ x.$$

Для транзакції, що «чіпляється» до z_1 та z_2 повинні виконуватись наступні умови:

1) комісія, котру вона повинна отримати, повинна повністю покриватися комісіями всіх її «предків»;

2) комісії z_1 та z_2 мають бути ненульові.

3) комісія, котру бере x, має бути пропорційна її PoW.

Автори вводять поняття збіжності множини транзакцій. Нехай, S – множина транзакцій. Тоді S збігається, якщо

$$\exists t \forall s \in S : t \in future(s). \tag{2.2}$$

Збіжність множини транзакцій Blockchain-free доводитсья у публікації протоколу [5]. Це поняття є надзвичайно важливим, для обґрунтування стійкості та масштабованності.

Визначення 2.5. . Наведемо визначення *ваги* транзакції з роботи [5]. Нехай P – $T - POSET$, нехай $x \in P$, тоді

$$Weight(x) = \sum_{i=0}^{|future(x)|} Work(y_i) \tag{2.3}$$

Кожна транзакція x публікує винагороду таку, що $Fee(x) > 0$, таку, що вона компенсує розподілену вартість передачі та верифікації транзакції.

Визначення 2.6. *Ціною транзакції* [5] x будемо називати загальну винагороду x і всіх її «предків» доступну для усіх її «нащадків».

$$Prize_P(x) = \sum_{y \in past(x)} fee(y). \tag{2.4}$$

Таким чином $Prize_P(x)$ є змінною величиною що максимальна на момент коли транзакція x не має нащадків й монотонно зменшується з ростом графу P. Ціна транзакції зростає від предків до нащадків. Дана величина є важливою характеристикою для майнера, так як показує йому максимальний розмір комісії, яку він може отримати після виконання PoW

для даної транзакції.

Розробниками криптовалюти було виставлено наступні вимоги до транзакцій:

1) Стимулювання майнерів працювати з більш свіжими транзакціями.

2) Направлена обробка транзакцій: намагання, щоб ціна вузла графчейну якумога швидше ставала нульовою і таким чином транзакція і всі її предки були видалені з динамічної стуктури даних, в якій містяться транзакції з невичерпаними Fee.
Fee має вичерпуватися по мірі збільшення кількості блоків підтвердження.

Визначення 2.7. Введено *відношення порядку вичерпання комісії* [5]: транзакція x вважається старшою за транзакцію y тоді і тільки тоді, коли $Weight(x) < Weight(y)$. Дане відношення є відношенням повного порядку – це означає, що множина транзакцій – T-POSET.

Таким чином, при додаванні нової транзакції x, знаходиться найдавніший предок транзакції з невичерпанною комісією і від нього на шляху до x збирається уся наявна Fee необхідна для покриття ціни даної транзакції.

Визначення 2.8. *Валова плата (gross fee* або *fee) [5]* – кількість комісії за транзакцію, що доступна будь-якому нащадку.

Визначення 2.9. *Чиста плата (nett cost* або *cost)* [5] – кількість комісії, що доступна «нащадкам>, за винятком нової монети, що буде згенерована після додавання транзакції в графчейн.

$$Cost(x) = fee(x) - \sum_{y<\prec x} fee(y). \tag{2.5}$$

Визначення 2.10. $Drain_P(x)$ – частина $fee(x)$ в графі P, що вже «відійшла» до «нащадків» x [5]. Введемо величину $\delta_i(x)$ – частина $fee(x)$,

яка відійшла від x до її «нащадка» y [5].

$$Drain_P(x) = \sum_{y \in future(x)} \delta_i y. \tag{2.6}$$

Таким чином, маємо:

1) якщо $future(x) = 0$, то $Drain(x) = 0$

2) якщо з x забрали усе $fee(x)$, то $Drain(x) = fee(x)$.

Враховуючи визначення 2.10, перепишемо визначення 2.6 наступним чином [5]:

$$Prize_P(x) = (fee(x) - Drain_P(x)) + \sum_{z \in past(x)} (fee(z) - Drain_P(z)). \tag{2.7}$$

Визначення 2.11. *Ціна множини транзакцій* X – це сума цін всіх її елементів [5].

$$Prize_P(X) = \sum_{x \in X} Prize_P(x). \tag{2.8}$$

2.1.3 Випуск нових монет

Створення нових монет є результатом додавання валідних транзакцій до графчейну. Що більш важливо, це процесс збільшення загальної грошової маси, починаючи з її нульвого значення. «Викарбувані» монети надходять прямо до користувача, незалежно від «fee».

Визначення 2.12. Нові «монети» «карбуються», коли виникає нова транзакція. Користувач вибирає розмір «змагання» та оплачує собі певне значення його вартості. Це значення залежить від розміру доступних даних до закриття транзакції і визначається наступним чином [5]:

$$Mint(x) = f(Work(x)/Weight(x)) \times \sum_{y_i \prec x} Mint(y_i), \tag{2.9}$$

для деякої монотонної функції f, наприклад $f(x) = \alpha x$, для деякого константного системного параметру α.

Зауваження. Дана величина постійно змінюється після створення транзакції x. На момент створення $Mint(x) = 0$.

Процесс верифіцакії транзакції є навмисне схожим до верифікації криптовалют на основі блокчейну, а є декілька відмінностей.

Всі користувачі беруть участь у процесі верифікації транзакції. Це означає, що всі користувачі мають збирати транзікції випущені іншими вузлами і записувати ті, що пройщ=шли процес верифікації. Також користувачі мають записувати валідні транзакції, аби потім надіслати їх новим учасникам процесу верифікації.

Користувачі верифікують транзакції при отриманні. При отриманні повідомлення про транзакцію x, користувач спочатку перевіряє, чи є «предки» транзакції прийнятними, потім перевіряє правильність виконаного PoW. Остання частина верифікації – перевірка валідності самої транзакції x. Формат транзакції має бути правильним (well-formed) й транзакція має бути легітимною – відповідати правилам даної криптовалюти.

Для подолання конфліктів необхідно ввести поняття *висоти* транзакції.

Визначення 2.13. Висотою ($Height$) транзакції x назвемо сумму роботи необхідної для виконання PoW всіх предків транзакції, та роботи для виконання PoW самої транзакції x [5]:

$$Height(x) = Work(x) + \sum_{z \in Past(x)} Work(z). \tag{2.10}$$

При конфлікті пропонується обирати «найвищу» тразакцію, а всі інші конфліктуючі транзакції слід вважати невалідними.

Слід відмітити деякі переваги та недоліки протоколу Blockchain-fre. Перспективним рішенням авторів протоколу стало введення залежності між кількістю монет в транзакції та об'ємом роботи, необхідним для

затвердження транзакції. Такий підхід дозволяє виконувати менше роботи, для підтвердження малих транзакції, що дає можливість використання криптовалют, заснованих на Blockchain-free для швидкої оплати невеликих покупок. Проте автори не навели детермінованого механізму обрахунку складості в залежності від розміру. Іншим недоліком є те, що автори не наводять оцінок часу прийняття транзакцій. Іншим недоліком протоколу є відсутність лінійного порядку блоків, адже протоколом передбачно лише частковий порядок на множині транзакцій.

2.2 GHOST

Як і у випадку Blockchain-free, автори GHOST [?] пооділяють думку щодо того, що найважливішим питанням Біткойна як технології є спроможність до масштабування та підвищення його пропускної спроможності. Розробники GHOST дослідили наслідки збільшення пропускної спроможності Біткойн на його захищеність від атак з подвійною витратою. Автори показали, що при високій пропускній спроможності злоумисник, потужність якого суттєво менша за 50% загальної обчислювальної потужності мережі, здатен здійснит атаку подвійної витрати [?].

Автори пропонують альтернативу під назвою GHOST правилу найдовшого ланцюга, який змінює процедуру вирішення конфліктів для блокчейну. Для вирішення конфліктів на кожному форці в ланцюзі GHOST обирає піддерево з найбільшою кількістью вершин у ньому, як мейнчейн. Ця модифікація протоколу полегшує вищезазначену проблему безпеки та допоможе у розвитку протоколам на основі блокчейну. Варіант GHOST був прийнятий та впроваджений проектом Ethereum [?], платформою розподілених додатків другого покоління, яка останнім часом привертає велику увагу. Для найкращого використання ємності блокчейну в ідеалі слід поєднувати всі рішення. GHOST, може

розглядатися як модифікація, яка дозволяє збільшити зобов'язання блокового ланцюга протоколу, що, в свою чергу, дозволить здійснювати більше транзакцій за менших витрат.

Важливо зазначити, що окрім зниження стійкості до атаки подвійної витрати, з підвищенням швидкості додавання нових транзакцій з'являється ще декілька недоліків: По-перше, майнери, які мають краще з'єднання з мережею, отримують більшу частку від загальної винагороди, за їхню частку в обчислювальній потужності. По-друге, стратегія егоїстичного майнінгу, який досліджували Еял та Сірер [?], може бути застосована слабшими майнерами. Ці проблеми залишаються невирішеними протоколом GHOST як таким, проте супутньому документі [?] автори досліджували додаткову модифікацію (сумісну з GHOST), що знижує перевагу майнерів з кращим з'єднанням та забезпечують додаткове збільшення пропускної спроможності.

2.2.1 Альтернативний погляд на модель Біткойн

Автори моделюють мережу Біткойн як орієнтований ациклічний граф $G = (V, E)$. Кожен вузол v має деяку частку $p_v \geq 0$ обчислювальної потужності всієї мережі: $\sum_{v \in V} p_v = 1$. Кожен окремий вузол v в мережі генерує блоки відповідно до Пуассонівського процесу зі швидкістю $p_v \cdot \lambda$, так що вся мережа генерує блоки відповідно до Пуассонівського процесу зі швидкістю λ (поточне значення протоколу ($\lambda = \frac{1}{600}$) було обрано Сатоші при створенні Біткойн) [10].

У контексті атакованої мережі використовується $\lambda = \lambda_h$ як швидкість створення блоку мережею чесних майнерів [10]. Швидкість нападника позначається $\lambda_m = q \cdot \lambda_h > 0$, для деяких $0 < q < 1$ [10]. На відміну від чесної мережі, автори припускають, що зловмисник ефективно створює довгі ланцюжки.

Визначення 2.14. Будемо називати «батьком» блоку x блок, вказівник на який міститься в x. Процес зміни структури дерева з часом обумовлено вибором батьківского блоку блоком, що генерується, – *політику вибору батьківського блоку*. Формально цей вибір автори моделюють як функцію $s(\cdot)$, яка відображає дерево блоків $T = (V_T, E_T)$ на блок $B \in V_T$, який повинен бути батьком наступного блоку. Кожен вузол може мати різне представлення дерева (він може знати про всі створені блоки) і, таким чином, застосовує s до стану дерева, що спостерігається вузлом.

Визначення 2.15. Для кожного блоку B позначаємо часом $time(B)$ його *(абсолютний) час створення*. Блоки, по суті, утворюють структуру дерева, що розвивається в часі, перший блок (блок генезис) коренем дерева, створений в момент запуску мережі Біткойн [3]; ми позначимо структуру цього дерева в момент часу як $tree(t)$, а $subtree(B)$ піддерево укоріненого на . Нарешті, глибина блоку B у дереві будемо позначати $depth(B)$ [10].

Визначення 2.16. Протокол Біткойну передбачає додавання вузлами мережі нових блоків на кінці *найдовшого ланцюга блоків* відомого цим вузлам в момент генерації. Відповідно, $longest(t)$ – найглибший листок дерева $tree(t)$.

Визначення 2.17. Термін «*основний ланцюг*» відповідатиме шляху від блоку генезису до листа, який обирається для розширення (зазвичай $longest(t)$). Основний ланцюг вважається вузлами єдиною прийнятою версією історії транзакцій.

Визначення 2.18. *Швидкість росту* основного ланцюга є одним із основних показників продуктивності системи, позначається β [10]. Формально час, за який кількість блоків у головному ланцюзі збільшиться з $n-1$ до n, – випадкова величина, позначається як τ [10]:

$$\tau = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \tau_i \tag{2.11}$$

Визначення 2.19. Тоді, β – швидкість додавання блоку до основного ланцюга [10]:

$$\beta = \frac{1}{E[\tau]} \tag{2.12}$$

Визначення 2.20. Ще один параметр, що вбудовано у протокол, – це *максимальний розмір блоку* [10] (у КБ), позначений b. У документі припускається, що існує великий попит на обробку транзакцій і що блоки завжди наповнені межами.

Визначення 2.21. Основним показником масштабованості Біткойн автори вважають кількість транзакцій в секунду – *Transaction Per Second* (TPS), яку система додає в основний ланцюг. Позначимо через K середню кількість транзакцій на КБ [10]. Тоді TPS [10]:

$$TPS(\lambda, b) := \beta(\lambda, b) \cdot b \cdot K. \tag{2.13}$$

2.2.2 Протоколи майнингу та консенсусу GHOST

Основною відмінністю протоколу GHOST від протокол Біткойн є нова політика вибору основного ланцюга в блоковому дереві. Як стверждують автори запропонована зміна в протокол підтримує поріг безпеки до атак 50% навіть якщо мережа страждає від екстремальних затримок, а зловмисник – ні. Це дозволяє дизайнерам протоколу встановлювати високі швидкості створення блоків та великі розміри блоків, не боячись наблизитись до 50-відсоткової межі, що, в свою чергу, означає, що можна підтримувати високу швидкість додавання транзакцій.

Формально протокол GHOST - це нова політика вибору батьків для блоку B, що додається. Ця політика переосмислює поняття основного ланцюга.

Для блоку B у дереві блоків T, нехай, $subtree(B)$ є піддеревом з

коренем B, $Children_T(B)$ – це множина блоків, що безпосередньо посилаються на B як на їх батьківський блок. Позначимо через $GHOST(T)$ політику вибору батьків, яку пропонують автори протоколу, визначену як результат наступного алгоритму [10].

Алгоритм 2.1. *Greedy Heaviest-Observed Sub-Tree (GHOST)*

Вхід: дерево блоків T

1) *set* $B \leftarrow$ *Genesis Block*
2) *if* $Children(B) = \varnothing$ *then return(B) and exit*
3) *else update* $B \leftarrow \underset{C \in Children_T(B)}{\operatorname{argmax}} \mid subtree_T(C) \mid$
4) *goto line 2*

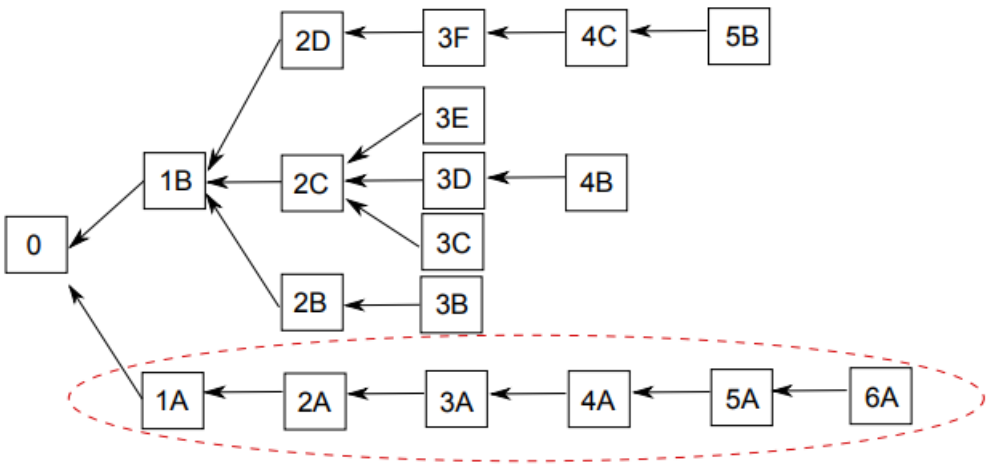

Рисунок 2.1 – Відмінність у виборі основної гілки між протоколами Біткойн та GHOST

Рисунок 2.1 ілюструє сценарій, за допомогою якого чесна мережа створювала дерево блоків з великою кількістю форків. Зловмисник таємно створює ланцюжок з 6 блоків (позначені 1, 2, ..., 6), який явно довший, ніж найдовший ланцюг мережі (закінчується блоком 5). Якби розповсюдження блоку по мережі було швидшим (по відношенню до швидкості створення), усі блоки в дереві чесної мережі формували б один довгий ланцюг і нападник не мав би можливості згенерувати ланцюг довший за ланцюг чесних майнерів.

Основна відмінність протоколу, полягає в тому, що блоки, які не входять до основного ланцюга (за версією консенсусу з Біткойн), можуть враховуватися під час підрахунку його ваги. Так, блок 1B є батьківським для $2B$, $2C$ і $2D$, які підтверджують його безпосередньо, і включають його у свій ланцюг. Аналогічно, блоки $3C$, $3D$ і $3E$ підтверджують і $1B$, і $2C$ як частину свого ланцюга. Протокол найважчого піддерева, що запропоновано у [10], використовує це і додає додаткової ваги блокам, допомагаючи гарантувати, що вони будуть частиною основного ланцюга.

Алгоритм проходить шлях від кореня дерева (генезис блок) і на кожному форці вибирає блок, що веде до найважчого піддерева. Наприклад, у дереві, зображеному на рисунку 2.1 піддерево блоку $1B$ містить 12 блоків, тоді як в 1A міститься лише 6. Алгоритм, таким чином, вибере $1B$ той, що входить до основного ланцюга, і перейде до розв'язання конфліктів всередині $subtree(1)$. Це призведе до вибору блоків 0, $1B$, $2C$, $3D$, $4B$ як основного ланцюга (а не найдовшого ланцюга, що закінчується блоком 5B). Завдяки цьому форки всередині $subtree(1B)$ не вносять жодних негативних наслідків для ваги $1B$, а навпаки – кожне додавання блоку до $subtree(1B)$ ускладнює його опускання з основного ланцюга. Зокрема, коли зловмисник публікує свій 6-блоковий довгий секретний ланцюг, ті ж блоки, що і раніше, залишаються в основному ланцюзі.

2.2.3 Деталі імплементації GHOST

Нижче наведено деякі додаткові подробиці щодо використання та реалізації правила вибору ланцюга GHOST [10].

Посилання на кілька батьків. Оскільки протокол вимагає знання блоків поза ланцюга всіма вузлами, пропонується, щоб їх заголовки поширювати на всі вузли (але не обов'язково цілі блоки). Інформація про блоки поза ланцюгом може бути додана як метадані в блоки, що

генеруються.

Розгортання. При низькій швидкості генерації блоків і при малих розмірах блоків GHOST [10] і звичайне правило найдовшої ланцюга поводяться однаково: всі блоки просто будуть на одному довгому ланцюзі. Відмінності між двома правилами з'являються лише при підвищенні пропускної спроможності. Тому прийняття GHOST [10] може бути поступовим при низьких швидкостях транзакцій - вузли будуть частково сумісні з версією найдовшої ланцюга до тих пір, поки швидкість додавання транзакцій не зросте (додаткові посилання на заголовки блоків можна розміщувати всередині полів, які звичайний протокол на даний момент ігнорує, і так можна підтримувати зворотню сумісність). Однак, збільшення розміру блоку Біткойн або коефіцієнт створення блоку вимагає жорсткого форку в протоколі. Отже, щоб ці зміни відбулися, необхідне прийняття GHOST [10] більшістю майнерів.

Налаштування складності. Враховуючи потенційно складні зв'язки між швидкістю зростання основного ланцюга, швидкістю створення нових блоків, і тим фактом, що GHOST [10] більше залежить від загальної швидкості створення блоків, автори пропонують змінити спосіб автоматичного коригування складності. Замість того, щоб орієнтуватися на певну швидкість росту найдовшого ланцюга, тобто β (що є поточною стратегією Біткойн[3]), запропоновано, щоб загальна швидкість створення блоку зберігалася постійною (λ). Зауважте, що співвідношення між β і складністю дуже складне, і тому поточний механізм налаштування складності Біткойн[3] не буде працювати за високої швидкості генерації блоків.

Винагорода майнерів і майнінг нових блоків. Хоча GHOST використовує блоки поза основного ланцюга для захисту, автори вважають, що найкраще виділяти нові монети лише майнерам блоків, що знаходяться на головній ланцюжку, подібно до того, як працює правило найдовшого ланцюга сьогодні. Швидкість випуску нових монет можна регулювати незалежно від швидкості створення блоків (але дуже схожим

способом), регулюючи кількість випущених монет на блок з урахуванням вимірюваної кількості блоків у недавньому минулому (наприклад, у вікні в 2 тижні).

Перевагою даного протоколу є заявоена авторами збільшена швидкість приняття транзакцій у порівнянні з Біткойном [3]. Проте разючим недоліком специфікації протоколу [10] є відстутність оцінок часу прийняття транзакцій. Також недоліком протоколу є відсутність лінійного порядку блоків. Ще одним недоліком є збільшення імовірності виключення блоків з головного ланцюжка за умов великих затримок в мережі.

2.3 Tangle

На думку авторів протоколу Tangle, одним з помітних недоліків протоколу Біткойн є концепція винагороди майнерам за транзакції будь-якої вартості. Очевидно, що важливість мікроплатежів в індустрії IoT, яка активно розвивається, збільшуватиметься. Сплата винагороди майнеру, яка перевищує суму переводу в транзакції, здається нелогічною. Крім того, позбутися плати в інфраструктурі блокчейну непросто, оскільки вона служать стимулом для майнерів виконувати свою роботу. Це призводить до іншої проблеми з існуючими технологіями криптовалют – гетерогенної (неоднорідної) природи системи. Є два різних типи учасників, ті, хто видає транзакції, і ті, хто схвалює транзакції. Конструкція цієї системи створює неминучу дискримінацію деяких учасників, що, в свою чергу, створює конфлікти, які змушують усі елементи витрачати ресурси на вирішення конфліктів. Вищезазначені питання виправдовують пошук рішень, суттєво відмінних від технології блокчейн, основною для Біткойн та багатьох інших криптовалют.

2.3.1 Загальні терміни та позначення

У розглянутій роботі автори пропонують ще один підхід, що не застовує технологію блокчейн. Цей підхід реалізовано у криптовалюті iota [7], яка була розроблена спеціально для індустрії IoT. У роботі описуються загальні риси Tangle (від англ. клубок) та обговорюються проблеми, які виникають при спробі позбутися блокчейна з підтримкою розподіленої історії транзакцій. Конкретна реалізація протоколу iota не обговорюється.

Визначення 2.22. Взагалі криптовалюта на основі клубка працює наступним чином. Замість загального ланцюжка блоків існує DAG (як і в Blockchain-free [5]), який називається *Tangle*.

Визначення 2.23. Транзакції в межах протоколу називаються *сайтами*. Сайти створюються вузлами, і додаються до множини сайтів Tangle (множина вершин орієнтованого направленого ациклічного графу), що є головним реєстром для зберігання транзакцій.

Визначення 2.24. Множина ребер Tangle отримується наступним чином: коли надходить нова транзакція, вона повинна підтвердити дві попередні транзакції. Ці підтвердження представляються як ребра орієнтованого графу, як показано на рисунку 2.2. Таким чином транзакція A *підтверджує* транзакцію B, якщо вони мають спільне ребро у Tangle.

Визначення 2.25. Якщо між транзацією A та транзакцією B немає ребра, але існує шлях довжиною не менше двох від A до B, вважається що A *опосередковано підтверджує* B.

Визначення 2.26. Існує також «генезис» транзакція, яка затверджується прямо чи опосередковано усіма іншими транзакціями (рис 2.3).

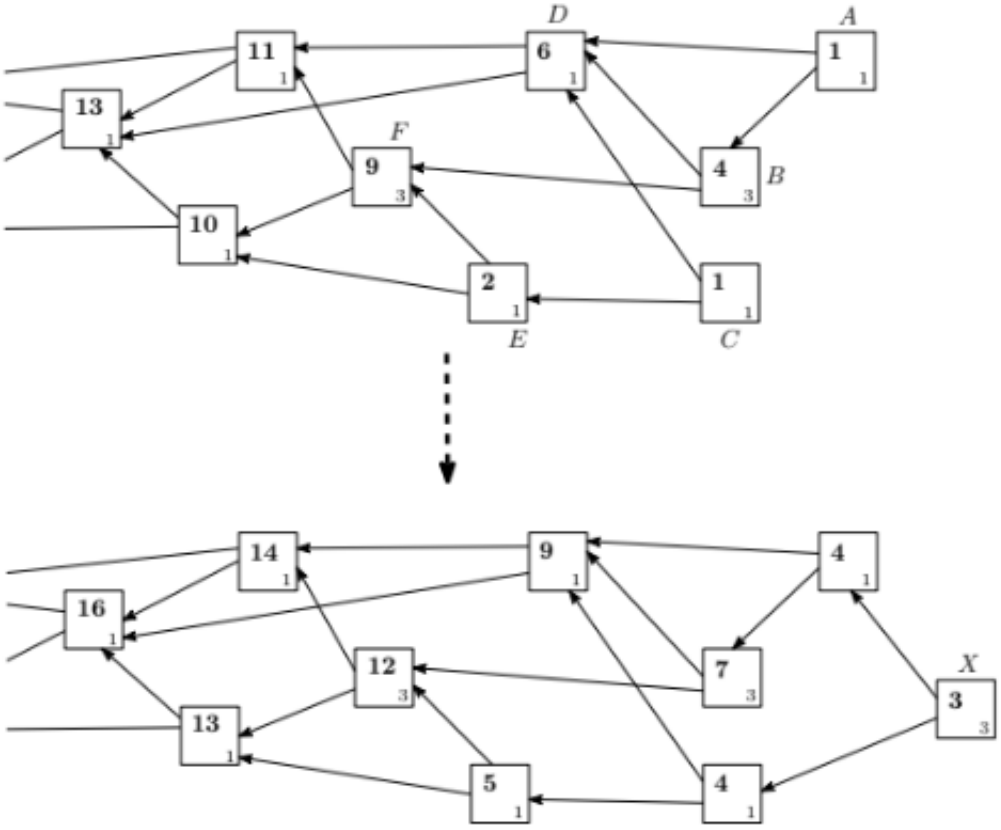

Рисунок 2.2 – Транзакція X валідує 2 попередні вершини дерева

Власне розвиток мережі описується наступним чином. На початку Tangle була адреса з рахунком, на якому знаходилися всі токени (аналог монет у Біткойні). Генезисна транзакція надіслала ці токени на кілька інших адрес «засновника». Підкреслимо, що всі токени були створені в генезис транзакції. Ніяких монет у майбутньому не буде створено, і майнінг не буде відбуватися в сенсі того, що майнери створюють монети при додаванні транзакції.

Зауваження. Основна ідея клубка полягає в наступному: щоб оформити транзакцію, користувачі повинні працювати над затвердженням інших транзакцій. Тому користувачі, які здійснюють транзакцію, вносять свій внесок у безпеку мережі. Передбачається, що вузли перевіряють, чи затверджені транзакції не суперечать один одному. Якщо вузол виявить, що транзакція суперечить історії розвитку клубка, вузол не буде затверджувати суперечливу транзакцію ні прямо, ні опосередковано.

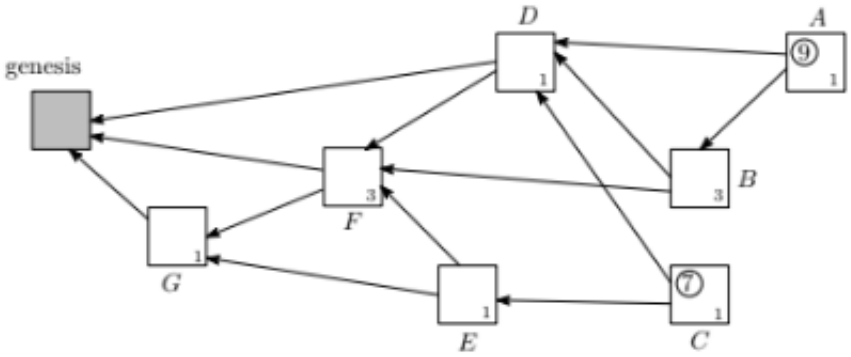

Рисунок 2.3 – DAG з генезис сайтом, власними вагами, присвоєними кожному сайту, та рахунками, розрахованими за сайти A і C

Для того, аби випустити транзакцію вузол має виконати наступні дії:

1) вузол обирає 2 транзакції для підтвердження відповідно до алгоритму. В загальному випадку ці дві транзакції можуть бути рівними одна одній.

2) вузол перевіряє чи ці дві транзакції не конфліктують. Конфліктуючі транзакції не підтверждуються.

3) для того, аби випустити валідну транзакції вузол має розв'язати криптографічну задачу, як в Біткойні. Це досягається знаходження нонсу такого, що геш від нонсу зконкатенованого з певними даними з підтверждених транзакцій має певний вигляд.

Зауваження. Слід відмітити, що мережа iota асинхронна. В загальному випадку, вузли не обов'язково бачать однаковий набір транзакцій. Слід також зазначити, що Tangle може містити суперечливі транзакції. Вузли не повинні досягти консенсусу щодо того, які дійсні транзакції мають право бути в головній книзі, тобто всі вони можуть знаходитись у клубі. Однак у випадку, коли є конфліктні транзакції, вузлам необхідно вирішити, які транзакції стануть сиротами. Основне правило, яке вузли використовують для вирішення між двома суперечливими транзакціями, полягає в наступному: вузол багато разів запускає алгоритм вибору підказок (tip selection algotithm або TSA) і

бачить, яка з двох транзакцій, швидше за все, буде опосередковано схвалена вибраною підказкою. Наприклад, якщо транзакцію було обрано 97 разів протягом 100 запусків алгоритму вибору підказок, ми говоримо, що це підтверджується з впевненістю у 97%.

Цікавим є механізм мотивування вузлів до розповсюдження транзакцій мережею. Кожен вузол обчислює деяку статистику, одна з яких - це кількість нових транзакцій, отриманих від сусіда. Якщо один конкретний вузол «занадто ледачий», сусіди відкинуть його з мережі. Тому, навіть якщо вузол не видає транзакції, а отже, не має прямих стимулів ділитися новими транзакціями, які затверджують його власну транзакцію, він все ще має опосердковані стимули брати участь у розпосюдженні транзакцій.

2.3.2 Вага та інші характеристики транзакції

Визначення 2.27. *Вага* транзакції x (позн. $w(x)$) – це деяка величина, що пропорційний обсягу роботи, яку в неї вклав вузол для підтвердження. У поточній реалізації iota вага може приймати лише значення 3^n, де n - натуральне ціле число, яке належить до деякого непорожнього інтервалу прийнятних значень. Насправді неважливо знати, як вагу отримували на практиці. Важливо лише, щоб кожна транзакція мала додатне ціле число, її вагу.

Визначення 2.28. *Кумулятивна (сукупна) вага* транзакції x (позн. $cw(x)$) – сума ваги конкретної транзакції сума ваг усіх транзакцій, які прямо чи опосередковано затверджують цю транзакцію (позн. $future(x)$).

На рисунку **??** зображено приклад Tangle DAG. Квадратами позначено транзакції. Число у нижньому правому куті позначає власну вагу транзакції, а число у лівому верхньому – сукупну вагу. Наприклад, транзакція F прямо або опосередковано затверджується транзакціями A,

B, C, E. Кумулятивна вага F дорівнює $9 = 3 + 1 + 3 + 1 + 1$, що є сумою власної ваги F і власні ваги A, B, C, E.

Визначення 2.29. *Верхівками* (англ. *tips*) називаються непідтверджені транзакції.

Визначення 2.30. *Висотою* транзакції x є довжина орієнтованого шляху до генезис транзакції (позначення $height(x)$).

Визначення 2.31. *Глибиною* транзакції x є довжина найдовшого обернено орієнтованого шляху до деякої верхівки (позначення $depth(x)$).

Пояснимо ці поняття на рисунку 2.2. Наприклад, $height(G) = 1$, а $depth(G) = 4$ через обернено орієнтований шлях F, D, B, A, тоді як D $height(D) = 3$, а $depth(D) = 2$.

Визначення 2.32. Нехай маємо транзакцію X. За визначенням, *рахунок* транзакції ($score(X)$) – це сума власної ваги всіх транзакцій, затверджених цією транзакцією плюс власна вага самої транзакції. На рисунку 2 2.2 - це A і C. Транзакція A прямо чи опосередковано затверджує транзакції B, D, F, G, тому оцінка A дорівнює $1 + 3 + 1 + 3 + 1 = 9$. Аналогічно, оцінка C дорівнює $1 + 1 + 1 + 3 + 1 = 7$.

Слід відмітити, що розробники Tangle умовно виокремлюють має 2 режими роботи мережі – режими низького та високого навантаження мережі. У режимі низького навантаження кількість tips є невеликою і часто рівна 1. Така ситауція складається за умови, коли кількість сайтів, що генеруються, настільки невелика, що стає малоімовірним сценарій за якого кілька транзакцій затверджують одну й ту tip. Також, якщо затримка мережі низька, а обчислювальна потужність девайсів висока, – малоімовірно, що у мережі буде підтримуватись велика кількість tips. Також автори протоколу вважають, що за цього режиму немає нападників, що підвищують кількість транзакцій. За режиму високого навантаження кількість генерування транзакцій висока, а затримки обчислень PoW та затримки мережі настільки високі, що стає можливим варіант, коли одну й ту саму tip підтверджують кілька транзакцій.

Автори протоколу стверджують, що даний протокол має більшу пропускну здатність ніж Bitcoin. Ще однією перевагою даного протоколу є те, що усі учасники мережі є майнерами. Головними недоліками є відсутність лінійного порядку на множині транзакцій та відсутність оцінок часу прийняття транзакцій.

Визначення 2.33. Нехай випадковий процес $\{L(t), t \in [0, \infty)\}$ може знаходитися в одному із станів $s \in S$, S – множина станів. Якщо ймовірність p повернення у будь-який стан рівна 1, то такий процес називається *рекурентним*. Якщо час повернення скінчений, то такий процес називається *позитивно-рекурентним*.

2.4 PHANTOM

У своїй роботі [?] автори представляють PHANTOM, протокол підтвердження транзакції, що задумувався як захищений від атак зловмисників за будь-якої пропускної спроможності, яку може підтримувати мережа. Протокол PHANTOM побудовано на основі орієнтованого ациклічного графу з блоків з транзакціями (англ. *blockDAG*), що є спільною рисою з Blockchain-free [5] та Tangle [7]. PHANTOM застосовує жадібний алгоритм на blockDAG для розрізнення між блоками, що були змайнені відповідно до протоколу чесними вузлами, та блоками, що були змайнені неузгодженими вузлами, що відхилилися від протоколу майнінгу. Використовуючи це розрізнення, PHANTOM вводить відношення повного порядку на blockDAG.

Одна з головною перевагою ФАНТОМ перед іншими, що для нього існує правило, яке дозволяє задати лінійний порядок на інших.

Протокол Біткойн вказує майнерам, як створювати блоки транзакцій. Тіло блоку містить записи про нові транзакції, опубліковані користувачами, доказ виконаного PoW та вказівник на один батьківський

блок. Останнє правило передбачає, що блоки природньо формують деревовидну структуру. Створюючи свій наступний блок, майнер обирає верхівку найдовшого ланцюга в дереві та ігнорує решту блоків (так звані покинуті (англ. orphaned) блоки).

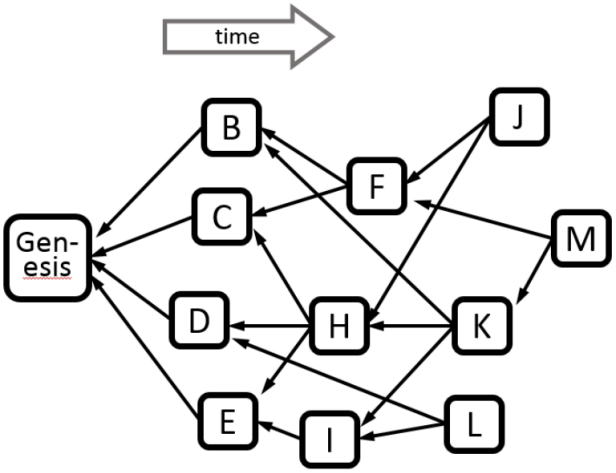

Рисунок 2.4 – Приклад дерева побудованого за протоколом PHANTOM.

Майнери розповсюджують блок відразу після отримання з інших вузлів або генерації та посилаються на останній блок у найдовшому для них в певний момент часу ланцюзі. Захищеність Біткойн залежить від того, щоб чесні вузли мали гарне мережеве з'єднання, так щою коли один майнер подовжує ланцюг новим блоком, ланцю вчасно передається на всі чесні вузли до того, як буде створено наступний блок. Щоб гарантувати цю властивість, створення блоків відбувається один раз на 10 хвилин, що передбачено протоколом. В результаті Біткойн страждає від дуже обмежувальної пропускної спроможності в порядку 3-7 транзакцій в секунду (tps). Замість того, щоб розширювати один ланцюг, манери в PHANTOM отримують інструкцію посилатися на всі блоки в графіку (на які раніше не було посилань, тобто блоків листів). Приклад blockDAG наведено на рисунку 2.4.

Основним завданням протоколу DAG є те, як впорядкувати транзакції що в ньому знаходяться, так що у випадку двох (або більше) суперечливих транзакцій приймається, та що надійшла першою

(відповідно до встановленого порядку), інша відкидається.

З цією метою PHANTOM покладається на зв'язанність чесних вузлів (подібно до припущення Біткойн у низькій пропускній спроможності). Оскільки співпрацюючі, майнери PHANTOM розповсюджують свої блоки якнайшвидше та посилаються блоки інших чесних майнерів. Очікується, що в DAG буде добре пов'язаний кластер блоків. На відміну від цього, блоки, що майняться вузлами, що не співпрацюють, будуть виглядати як застарілі та легко розпізнаватися. Дійсно, відхилення від протоколу майнінгу PHANTOM відбувається у вигляді утримування нового блоку на деякий час, та
або створення нового блоку, що не посилається на жоден з існуючих блоків. Обидва випадки відстежуються за допомогою протоколу. Згенеровані таким чином блоки відкидаються.

2.4.1 Необхідні терміни та позначення

Головним поняттям протоколу є знайоме нам поняття DAG. Нехай G – DAG, $G = (C, E)$, де C – можина вершин, E – множина ребер. Також у протоколі використовуються такі поняття як $past$, $future$, $tips$.

Визначення 2.34. Розглянемо DAG G з транзакціями. Для деякого блоку B, $B \in C_G$, $anticone(X)$ – множина блоків з G, відношення між якими та блоком B є невизначеними. Таким чином:

$$anticone(X) = C_G \setminus (past(X) \cup future(X) \cup \{X\}). \tag{2.14}$$

Ще одне можливе визначення $anticone(B)$ – це множина блоків, що не вказують на B прямо або опорсередковано. На рисунку 2.4 $anticone(H) = B, F, I, L$.

Головною задачею з теорії складності обчислень, на якій базується стійкість протоколу є задача пошуку максимального k-кластеного

орієнтованого направленого підграфу (Maximum k-cluster SubDAG або MCS_k). В цій задачі на вхід подається DAG $G = (C, E)$. На виході отримуємо підмножину $S^* \subset C$ максимальної потужності, тобто такої, що

$$\forall B \in S^* : |anticone(B) \cap S^*| \leq k.$$

Визначення 2.35. Нехай дано DAG $G = (C, E)$, множина $S \subseteq C$ – *k-кластер*, якщо $\forall B \in S : |anticone(B) \cup S| \leq k$

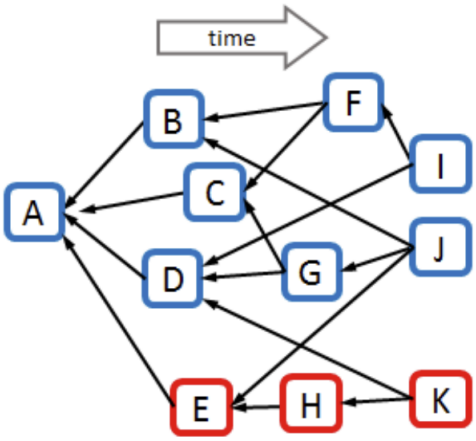

Рисунок 2.5 – Приклад найбільшого 3-кластеру

На рисунку 2.5 синім кольором позначено максимальний 3-кластер в DAG: A, B, C, D, F, G, I, J. Неважко перевірити, що кожен із цих синіх блоків має щонайбільше 3 блоки у своєму *anticone*, і, що це найбільший набір із цією властивістю. Встановлення параметру взаємозв'язку PHANTOM з k = 3 означає, що максимум 4 блоки створюються в межах кожної одиниці затримки, так що розміри *anticone* не повинні перевищувати 3. Блоки поза найбільшим 3-кластером позначено червоним: E, H, K, створені зловмисником. Наприклад, блок E має 6 синіх блоків у своєму *anticone* (B, C, D, F, G, I); ці блоки не посилалися на E. Аналогічно, блок K також має 6 синіх блоків у своєму антиконі (B, C, F, G, I, J); Імовірно, його нечесний майнер отримав уже деякі блоки з (B, C, D, G), але порушив протокол майнінгу, не посилаючись на них.

Насправді, задача MCS_k NP-складною [?]. Тому PHANTOM працює з варіантом цієї задачі, використовуючи жадібний алгоритм пошуку. Алгоритм наведено у розділах 2 та 6 специфікації протоколу [?].

В загальному вигляді протокол PHANTOM складається с наступних трох кроків:

1) використовуючи структуру DAG знайти максимальний k-кластер – розпізнати гарно з'єднані блоки. Як стверджують автори протоколу це є, з великою ймовірністю, блоки, що були згенеровані чесними майнерами.

2) пронумерувати блоки в графі в залежності від їх приналежності до k-кластеру.

3) пронумерувати транзакції в блоці.

Алгоритм 2.2. Наведемо приблизний вигляд алгоритму побудови k-кластеру.

1) для кожного *tip* x побудувати $past(x)$. З отриманої множини $past(x)$ вибрати максимальний k-кластер.

2) вибрати найбільший k-кластер з усіх *tips*.

3) розширити його, перевіряючи (в певному порядку) блоки, що в нього не потрапили.

Інтуїтивно, спочатку вибирається вершина DAG з найбільшим *score* – B_{max}, де $score(B) = |BLUE_k(past(B))|$. Далі починається процес «фарбування» блоків з $anticone(B_{max})$, таким чином, щоб зберігалася власивість k-кластерую. Власне кажучи, цей алгоритм діє, як алгорит вибору гілки графу: B_{max} – вершина гілки графу, потім обирається вершина з найбільшим *score* в $past(B_{max})$ – її предок в гілці і так далі. Отримана гілка позначається як $Chn(G) = (genesis = Chn_0(G), Chn_1(G), ..., Ghn_h(G))$. Автори протоколу такою поведінкою обраховують апроксимацію розв'язку задачі пошуку максимального k-кластеру в піддереві.

Визначення 2.36. Нехай G – DAG. Тоді блакитна множина

$Blue_k(G)$ – максимальний k-кластер, що отримується на виході алгоритму побудови 2.2.

Наступне співвідношення є вірними для $Blue_k(G)$:

$$\forall B \in G : score(B) = |past(B) \cap Blue_k(G)| = |Blue_k(past(B))|.$$

Тобто $score(B)$ – це кількість блакитних блоків в $past(B)$.

Визначення 2.37. B_{max} – tip з максимальним $score(tip)$.

Нехай $G = G^{pub}_{\infty}$ – усі блоки; $B \in G$. Тоді виконується:

1) якщо B створено чесними майнерами, тоді ймовірність $P(B \notin Blue_k(G))$ експоненційно спадає зі збільшенням k;

2) якщо B створено нечесними майнерами, тоді ймовірність $P(B \in Blue_k(G))$ експоненційно спадає зі збільшенням T, де T – час який блок B не розповсюджувався по мережі.

Алгоритм 2.3. Алгоритм нумерації блоків в $Blue_k(G)$ має наступний вигляд:

1) проініціалізувати чергу множин пустою множиною;

2) обійти всі вершини в $Blue_k(G)$, що знаходяться в деякому частковому порядку обумовленому топологією DAG;

а) для кожного блоку B спочатку додавати в чергу вершини з $past(B)$;

Зауваження. Нехай $B \in Blue_k(G)$, $C \notin Blue_k(G)$. Тоді:

$$C \prec B \Longrightarrow \exists D \in Blue_k(G) : C \in past(D)$$

Тобто блок з $AntiBlue_k(G)$ буде пронумеровано тоді, коли дошла черга нумерації якого-небудь блакитного блоку з його $future$.

2.4.2 Майнінг

Майнінг в межах протоколу PHANTOM відбувається за правилом Proof-of-Work. Нові блоки додаються за Пуассонівським розподілом з параметром λ. Розглянемо деякого майнера $v \in N$, де N – множина усіх майнерів. Тоді v володіє деякою часткою α_v загальної обчислювальної потужності. Ще α_v – імовірність того, що наступний блок буде згенеровано саме майнером v. Нехай $malisious$ – множина нечесних майнерів, $Honest$ – множина чесних майнерів. Тоді за припущенням авторів протоколу у мережі має виконуватись наступне твердження: $\sum_{v \in malisious} \alpha_v < 0.5$. Також введемо позначення для стану DAG в певний момент часу t: G_t, DAG, спостережуваний чесними майнерами – G_t^{pub}.

Протокол майнінгу має наступний вигляд:

1) як тільки вузол створює або отримує новий блок, він передає його на всі інші відомі вузлу піри в N:

$$\forall u, v \in honest : G_t^v \subset C_{t+D}^u$$

2) при створенні нового блоку B майнер v має додати у заголовок цього блоку геші усіх предків цього блоку:

$$v \in honest \Longleftarrow past(B) = G_t^v$$

Для впорядкованого DAG G виконується:

1) B створено раніше $C \iff B \prec_{ord(G)} C \iff B \prec_G C \iff$
$\iff B \prec C$

2) якщо $C \notin G$, то все одно $B \prec C$.

Визначення 2.38. Нехай $B \in G = G_{t_0}^{pub}, t_1 \geqslant t_0$, тоді функція ризику $Risk(B, t_1)$ має наступний вигляд:

$$Risk(B, t_1) = Pr(\exists s > t_1, \exists C \in C_S^{pub} : B \prec_{G_S^{pub}} C \wedge C \prec_{Gn_S^{pub}} B)$$

Твердження 2.1. *Правило впорядкування ord збігається з часом навіть якщо існує деяке частка нечесних манерів* α*:*

$$\forall t_0 > 0, \forall B \in G_{t_0}^{pub} : \lim_{t_1 \leftarrow \infty} Risk(B, t_1) = 0.$$

Визначення 2.39. *Поріг безпеки* деякого правила впорядкування *ord* – це максимальне значення α (частки обчислювальної потужності атакуючої сторони), для якого виконується твердження 2.1.

Визначення 2.40. Блок B – є «пісчаним годинником» (англ. Hourglass block), якщо виконується наступне співвідношення:

$$Blue_k(G) \cup anticone(\hat{B}) = \varnothing,$$

що є тотожнім до одночасного виконання наступних умов:

1) з усі блоків з $Blue_k(G)$ існує орієнтований чи оберенено орієнтований шлях до B;

2) не існує блоків з яких нема орієнтованого чи оберенено орієнтованого шляху до B.

Перевагою протоколу PHANTOM над іншими розглянутими протоколами є наявність лінійної упорядкованості блоків. Щоправда алгоритм вибору валідних блоків може зациклюватися при перевантаження мережі. В цьому випадку сині транзакції можуть ставати червоними. Ще однією перевагою даного протоколу є наявність оцінок часу прийняття блоків. Щоправда, дані оцінки є вкрай неточними.

Висновки до розділу 2

У даному розділі розглянуто деякі протоколи на графчейнах. Представлено основні принципи побудови їх структури та висвітлено їх переваги та деякі недоліки. До переваг слід віднести відносно невеликий

PoW, що дає можливість майнерам успішно генерувати блоки/транзакції без необхідності об'єднуватися в величезні майнінгові пули. Нажаль у розглянутих протоколах є значні недоліки. По-перше, всі протоколи окрім PHANTOM не мають лінійого порядку на множині блоків/транзакцій. Оцінки часу прийняття транзакцій у всіх протоколів або відстутні, або надзвичайно неточні.

3 ВРАЗЛИВОСТІ ПРОТОКОЛІВ, ЗАСНОВАНИХ НА ТЕХНОЛОГІЇ ГРАФЧЕЙН

У цьому розділі ми розглядаємо дві моделі атак на протоколи, що засновані на технології блокчейн (проте застосовані підходи класифікації атак справедливі і для протоколах на графчейн). Після розгляду видів атак у цій главі ми переходимо до аналізу вразливостей до цих атака протоколів, які базуються на графчейнах.

3.1 Основні атаки на блокчейн та графчейн

У літературі можна знайти 2 види атак на Бійткой та блокчейн взагалі. У даній роботі нами було побудовано атаку подвійної витрати на протокол Blockchain-free, заснований на технології графчейн, тим самим, ми показали, що один із видів атак, про який буде вестись мова далі, є дійсним і для узагальнення блокчейну в технології графчейн.

3.1.1 Атака подвійної витрати

Цей тип атаки зустрічається у багатьох публікаціях включаючи [14, 16, **?**] та ін. Зазвичай вона має назву «Атака подвійної витрати». Атака відбувається наступним чином. Противник здійснює деяку транзакцію в блоці з номером N, передаючи монети постачальнику товарів за деяку покупку. Постачальник отримує необхідну суму монет і відповідно постачає товар покупцеві. Отримавши товар, супротивник швидко починає видобуток іншого блоку з тим же номером N, тобто блоку, що слідує за блоком $N-1$, але такого, який або не містить цієї

транзакції, або він перераховує гроші собі на іншу біткойн-адресу. А щоб гарантувати прийняття цього альтернативного ланцюга чесними майнерами, він намагається підчепити якомога більше блоків до альтернативного блоку N. Якщо йому вдасться зробити альтернативну ланцюжок довше ніж ланцюг, що містить блок з транзакцією оплати товарів, то саме цей ланцюг, згідно з протоколом майнінгу, буде прийнято до головного ланцюжка. Очевидно, що чим більша частка, яку утримує противник (не важливо, чи це обчислювальна потужність у випадку PoW, чи частка від монет у випадку PoS), тим більше шанс його атаки бути успішним. Зокрема, якщо частка противника перевищує $\frac{1}{2}$, то ймовірність успіху атаки дорівнює 1.

Щоб забезпечити захист від цієї атаки, Накамото у своєму першому документі [3] запропонував не поставляти товари, як тільки відбулася транракція, а почекати деякий час, точніше – генерації деякої кількості блоків після цієї транзакції, і лише тоді, якщо транзакція не зникла з блокчейну, поставити товар. У цьому випадку супротивник не може побудувати форк відразу після платежу, оскільки тоді продавець побачить, що транзакція то зникає, то з'являється в блокчейні, і відхилить транзакцію. З цієї причини супротивник спочатку чекає, поки над блоком з транзакцією «виростає» за необхідна кількість блоків підтвердження. Протягом цього періоду очікування він може непомітно генерувати форк з блоками перед блоком з транзакцією, тобто, в наших позначеннях, може генерувати альтернативний блок N з наступними блоками, але ні в якому разі він не поширює цей альтернативну ланцюг в мережу поки йду підтвердження, щоб продавець не запідозрив, що його атаковано. Це перша етап атаки. А коли блоки підтвердження сформовані і товар отриманий, противник намагається «наздогнати» існуючий ланцюг, і це вже друга стадія нападу. Припустимо, що, хоча генерується 6 блоків підтвердження, противник зміг генерувати 4 блоки альтернативного ланцюга. І зараз він відстає як мінімум на 2 блоки. Якщо колись у майбутньому він зможе генерувати стільки блоків, скільки

потрібно, щоб «наздогнати» існуючий ланцюжок, який, у свою чергу, також буде постійно зростати, тоді атака буде успішною. Зокрема, якщо йому вдалося генерувати 7 або більше блоків на першій стадії атаки, поки він чекав блоків підтвердження, то атака вже була успішною. Отримавши товар, він просто представляє власний довший ланцюжок, в якому гроші залишаються з ним.

3.1.2 Атака розгалуження

Цей тип атаки набагато менш відомий ніж атака подвійної витрати. Його було описано лише у двох працях [12, 13]. В даній атаці злоумисник намагається створити якомога довший форк, намагаючись розкласти свої блоки між двома гілками форку таким чином, щоб зберегти існування двох гілок однакової (або майже однакової) довжини. У той же час супротивник не приховує самого факту, що форк існує, і сторонній спостерігач може легко побачити, що головний ланцюг складається то з одних, то з інших блоків. Отже, деякі транзакції зникають, а потім з'являються. Чесні майнери, які також беруть участь у побудові блокчейну, повинні дотримуватися протоколу майнінгу, тому вони повинні подовжувати один ланцюг, а потім інший, тим самим мимоволі беручи участь у підтримці існування злоумисного форку.

Зауважте, що в Атака I немає цих двох етапів, вона має «видимий» форк протягом усього процесу. У цьому випадку, на відміну від Атаки I, виграш противника в Атаці II очевидний: він купував товари і не витрачав грошей. Але у випадку Attack I максимум, що він може зробити, - це скомпрометувати криптовалюту, і це не принесе йому ніякої користі; якщо йому також належить ця валюта, він навіть понесе збитки (обвал стоимости криптовалюті).

У роботі [12] було розглянуто даний тип атаки для PoW у моделях Біткойн та GHOST [10]. В роботі отримані наступні результати:

1) аналітична оцінка верхньої межі ймовірності форку довжини l з заданими m та n (частками відповідно нападника та чесних майнерів);

2) на базі цієї оцінки були отримані чисельні оцінки для частки $\frac{m}{m+n}$ обчислювальної потужності зловмисника.

Незважаючи на те, що отримані оцінки не виражені напряму в термінах експоненційних функцій, вони підходять для отримання чисельних результатів і побудови відповідних графіків. Вони показують, що ймовірність експоненційно спадає з ростом довжини форку.

3.2 Вразливість протоколу Blockchain-free та побудова атаки на цей протокол

Першим протоколом, вразливості якого будуть розглянуті, є протокол Blockchain-free [5]. У цій роботі ми знайшли багато суперечливих тверджень. Зараз ми перейдемо та обгрунтування недоліків, виявлених у твердженнях та термінології протоколу Blockchain-free [?].

3.2.1 Некоректність побудови Blockchain-free

Недолік 1

Наступне означення є означенням 1 в [5].

Визначення 3.1. *Схема proof-of-work* характеризується функцією S що приймає довільні рядки a, разом з деяким розв'язком рядком b, де $S(a, b)$ повертає 1 або 0. Нехай функція S має обчислювальну складність d, позначення $S = S_d$ та $Work(S) = d$, якщо хоча 1 раз можна отримати таке b, що $S(a, b') = 1$ за k випадкових $a \in A$, можна з імовірністю

$$Pr[S(a, b') = true] = kd^{-1} + negl(a), \tag{3.1}$$

таким чином формуючи функцію $negl(a)$.

Зауваження. В даному визначенні потрібно додати умову $k \ll d$. В іншому разі ймовірність буде більше 1.

.

Недолік 2

У роботі автори використовують визначення відношення порядку для транзакцій, яке звучить як наше 2.7.

Твердження 3.1. *Ми вважаємо, що дане твердження суперечить означенню 2.3, у [5] це означення 3.*

Доведення. Якщо $x \prec y$, тоді $future(x) \supset future(y)$, звідси отримуємо, що $Weight(x) \geq Weight(y)$. Отримуємо, що біль стара транзакція має більшу вагу, що є нонсенсом. □

Таким чином ми помічаємо суперечності в одному із базових понять в Blockchain-free. Тож усі твердження і доведення в даній роботі можна вважати, меншою мірою сумнівними.

Автори протоколу намагаються дати **оцінки часу виснаження винагороди за транзакцію**, що є темої глави 2.1 в [5]. Автори стверджують, що вони можуть контролювати час, необхідний для того аби ціна транзакції повністю вичерпалася й пропонують наступне співвідношення:

$$\frac{Time - to - drain(sec.)}{Age - of - system(sec)} = \beta \frac{Prize_P(P)}{\sum_{y_i \in P} Work(y_i)} \tag{3.2}$$

Як бачимо, в даному відношенні існує прив'язка по загальному часу, а воно може бути локальним. Також не зрозумілим залишається той факт, чому автори ввели це поняття для множини транзакцій, а не для окремих транзакцій. Співвідношення часу виснаження ціни окремої транзакції так і не було подано авторами.

З частини 2.1.2 статті [5] нашу уваги привернули 2 твердження. Перше стосується конфліктуючих транзакцій, друге – вибору більшістю конфліктних гілок графчейну.

Недолік 3

Автори протоколу пропонують наступну поведінку при конфлікті транзакцій: x та y перевагу має транзакція яка має за собою більшу кількість роботи $Work(x)$ чи $Work(y)$.

Зауваження. Таким чином, можна зробити висновок, що перевагу має більш пізня транзакція, що є дещо дивним, адже логічно припустити, що майнер, який зумів виконати необхідну для виконання PoW роботу першим, повинен мати перевагу над майнером, який виконав роботу другим.

Недолік 4

Автори стверджують, що перевіряючі, що приймають гілку конфлікту без консенсусу, незабаром будуть вимушені прийняти консенсус більшості, оскільки більшість подовжує графік із гілки консенсусу швидше, ніж дисиденти. Це гарантує, що гілка більшості належить або буде гілкою консенсусу.

Зауваження. Це твердження не було строго доведено у роботі. Можливі варіанти, коли твердження не буде вірним, наприклад якщо мережа буде страждати від великих затримок.

Недолік 4

Розробники протоколу припускають, що більшість учасників протоколу поводяться раціонально.

Зауваження. Розробники протоколу не дають визначення раціональної поведінки. Також вони не повідомлють в чому може заключатися раціональна поведінка. Навіть якщо раціональність заключається в більшому об'ємі можливої винагороди, яку можна отримати за майнінг, це не є захитом від атаки подвійної витрати.

3.2.2 Атака подвійної витрати на Blockchain-Free

За нашим припущенням, зловмисник має менше половини загальної обчислювальної потужності мережі, проте може робити прихований майнинг своїх транзакцій, «чіпляючись» (валідуючи) до кожної потрібної транзакції, та має повний доступ до інформації про стан DAG, як і чесні учасники.

Серед недоліків протоколу можна виокремити той факт, що автори явно не описали скільки попередніх транзакцій слід перевірити під час створенні нової.

Алгоритм підготовки атаки подвійної витрати:

1) зловмисник майнить своє піддерево з правильних транзакцій, але не відправляє їх у загальну мережу. Ці транзакції мають також посилатися на транзакції з відкритиго DAG чесних майнерів;

2) маючи достатню кількість транзакцій (з $Work$ більшою за ту, що потрібна для прийняття транзакції жертвою), зловмисник публікує першу транзакцію, пов'язану з публічним DAG, побудованим чесними майнерами, чекає, поки транзакція приймається жертвою – жертва надає послугу;

3) приєднує конфліктуючу транзакцію до таємно згенерованого піддерева і публікує його в мережу.

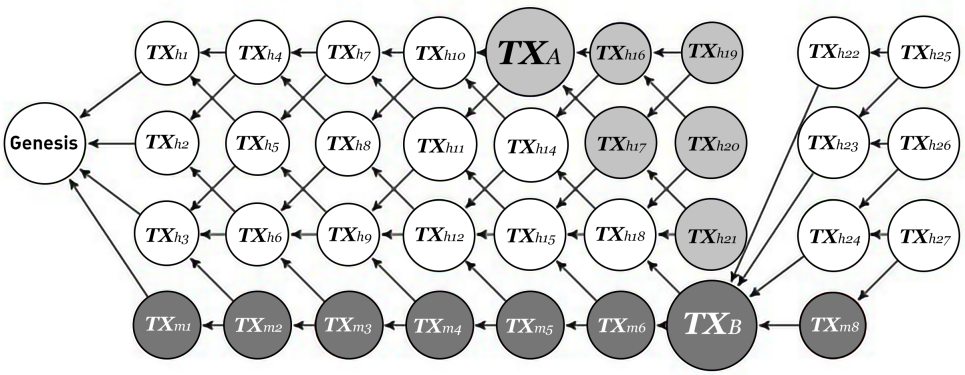

Рисунок 3.1 – Схема атаки на Blockchain-Free

На рисунку 3.1 наведено приклад атаки (на ньому не позначено 2 предки кожної транзакції, різний час генерації тощо). Також ми припускаємо, що кожна транзакція має однаковий PoW.

Білі транзакції згенеровані чесними учасниками і залишаються у DAG після атаки. Сірим кольором позначені транзакції, що згенеровані чесними учасниками, і транзакція TX_A, що генерується зловмисником. Вони будуть відкинуті в результаті атаки. Темно-сірі – це транзакції, що були згенеровані зловмисником. Його піддерево складається з таємно згенерованих $TX_{m1}, ..., TX_{m6}, TX_B$, що потім будуть опублікованими. TX_8 є опціональною транзакцією, що додається для підвищення ймовірності успіху (може не додаватись). TX_A й TX_B – конфліктуючі транзакції: у результаті атаки TX_A буде заміщено на TX_B.

Хід атаки у прикладі:

1) зловмисник приховано майнить правильно сформовані транзакції $TX_{m1}, ..., TX_{m6}$ до того моменту, поки він буде мати достатньо переваги у PoW.

2) зловмисник публікує транзакцію TX_A та очікує на блоки підтвердження TX_{16}, TX_{17}, TX_{19}, TX_{20}, TX_{21}, необхідні для підтвердження TX_A (кількість блоків підтвердження не визначається [5] та може варіюватися).

3) зловмисник генерує транзакцію TX_B, що конфліктує за TX_A у піддереві та публікує його.

4) чесні майнери бачать TX_B – найвищу правильно сформовану транзакцію з максимальним PoW в ній та в $Past(TX_B)$ і приймають її як валідну, відкидаючи підграф, що містить TX_A. Дійсно, підграф, що включає TX_B, складається з 23 транзакцій – 7 темно-сірих та 16 білих; підграф, що включає TX_A, містить 22 транзакції – 16 білих та 5 світло-сірих.

Теорема 3.1. *Нехай час, потрібний на виконання оплаченого замовлення відомий і дорівнює T. Нехай P_M – доля обчислювалльних ресурсів зосереджених у зловмисника, P_H – доля обчислювалльних*

ресурсів зосереджених у чесних майнерів, $P_M + P_H = 1$, α – *інтенсивність генерації блоків (блоків за секунду),* α_M – *інтенсивність генерації блоків зловмисником,* α_H – *інтенсивність генерації блоків чесними майнерами. Очевидно, що* $\alpha_H = P_H\alpha$, $\alpha_M = P_M\alpha$. *Тоді для часу підготовки атаки* T_attack *вірним є наступне співвідношення:*

$$T_{attack} \geq \frac{\alpha_H T + \sqrt{10^3 \alpha_H T} + 2z_0}{\alpha_M} \tag{3.3}$$

Доведення. Нехай P_A – пік (что? (підграф?) крайняя транзакція з найдільшою вагою, що містить транзакцію X_A), що містить X_A, P_B – пік (что?), що містить X_B. Нехай $Work(P_A) = Work_1 + Work_2 + Work_3$, де $Work_1$ – сумарна PoW транзакцій, які в $Past(X_A) \cap Past(X_B)$; $Work_2$ – сумарна PoW транзакцій, які злоумисник згенерував в $Future(X_A)$ до викладання; $Work_3$ – сумарна кількість роботи витраченої на підтвердження транзакцій, які згенерували чесні майнери після викладання X_A і до поставки товару. $Work(P_B) = Work_1 + Work_4 + Work_5$, де $Work_2$ – сумарна PoW транзакцій, які знаходяться в $Past(X_B)$ але не в $Past(X_A)$; $Work_3$ – сумарна PoW транзакцій, які згенерували чесні майнери після викладання X_B і до поставки товару. Очевидно, що для успішної атаки необхадно і достатньо, виконання наступного співвідношення:

$$Work_1 + Work_2 + Work_3 < Work_1 + Work_4 + Work_5.$$

Скоротивши $Work_1$, отримаємо:

$$Work_2 + Work_3 < Work_4 + Work_5.$$

Зазначимо, що зловмисник має необмежений час для генерації $Work_4$. Тому, навіть не враховуючи $Work_5$, достатньою умовою атаки буде

$$Work_2 + Work_3 \leq Work_5. \tag{3.4}$$

Для спрощення вважатимемо, що кожен блок має однакову кількість роботи w, тоді перепишемо нерівність 3.4:

$$N_1 + N_2 \leq N_4, \tag{3.5}$$

де N_i – кількість блоків, що відповідає $Work_i$:

$$N_i = \frac{Work_i}{w} \tag{3.6}$$

За час T чесні майнери в середньому створять $\alpha_H T$ блоків, тоді 3.5 матиме вигляд:

$$N_4 > N_2 + alpha_H T, \tag{3.7}$$

або, враховуючи дисперсію Пуассонівського процесу, що описує кількість створених блоків за час T, величину $\alpha_H T$ замінемо на $N = \alpha_H T + \sqrt{10^3 \alpha_H T}$. За нерівністю Чебишева **??**, з імовірністю не меншою за $1 - 10^{-3}$ чесні майнери за час T створять не більше N блоків. Для підтвердження транзакції потрібно $N_2 \geq z_0$. Покладемо $N_2 = z_0$; отримаємо

$$N_4 \geq \alpha_H T + \sqrt{10^3 \alpha_H T} + z_0. \tag{3.8}$$

Умова 3.8 є достатньою умовою виконання атаки з імовірністю не меншою за $1 - 10^3$ (ми можемо збільшити імовірність успіху шляхом збільшення N_4). Оцінимо час T_{attack}. Для її успішності зловмиснику потрібно створити $N_4 + z_0$ блоків, тобто не менше за $N_M = \alpha_H T + \sqrt{10^3 \alpha_H T} + 2z_0$ (почему 2 $2z_0$). Якщо інтенсивність генерації блоків злоумисником дорівнює α_H, то маємо наступне:

$$T_{attack} \geq \frac{N_M}{\alpha_H} = \frac{\alpha_H T + \sqrt{10^3 \alpha_H T} + 2z_0}{\alpha_M},$$

що доводить дану теорему. □

3.3 Недоліки протоколу PHANTOM

Другим протоколом який ми аналізували був PHANTOM [9]. В ньому також було знайдено багато недоліків, тому заявлена стійкість та ефективність не є доведеною. Зараз ми перейдемо та обгрунтування недоліків, виявлених у твердженнях та термінології протоколу Blockchain-free [9].

3.3.1 Некоректність побудови протоколу PHANTOM

Протокол здається доволі простим і його функціональність видається зрозумілою, оскільки на одному простому правилі. Необхідні для цього леми доведено коректно. Незважаючи на це нами було знайдено помилки у твердженнях з роботи.

В лемі 9 з [9] приводиться позначення події $\mathcal{E}(t_0)$ як такої, що володії наступними властивостями:

1) Деякий блок $\hat{B}$ було створено в деякий момент часу $u > t_0$ чесним вузлом () і окрім $\hat{B}$ не було жодного створено жодного блоку в інтервалі $[u - D, u + D]$;

2) Для декого $_1$ k найпізніших блоків в множині $BLUE_k(past(\hat{(}B)))$, позначимо $LAST_k(past(\hat{(}B)))$, було створено в період часу $[u - T_1, u]$ і нечесними майнерами не було створено жодного блоку в цьому інтервалі.

3) рахунок гілки в якій міститься $\hat{B}$ завжди більший за рахунок будь-якої гілки, що не містить $\hat{B}$: $\forall s \geq u$, $\forall C_1, C_2 \in tips(G_s^{pub}) : score(C_1) \geq score(C_2) \Longrightarrow \hat{B} \in Chn(past(C_1))$.
Це означення використовується у кількох твердженнях з [9] в яких ми знайшли недоліки у ході аналізу.

Недолік 1

Твердження 3.2. *Дане твердження є твердженням 1 в*

роботі [9]. Нехай блок $\hat{B}$ володіє першими двома властивостями блоку $\mathcal{E}(t_0)$, тоді поки $\hat{B}$ синій, $\hat{B}$ знаходить у минулому усіх інших синіх блоків: $\forall s \geq u, \forall B \in anticone(\hat{B}) : B \notin BLUE_k(G_s^{pub}) \vee \hat{B} \notin BLUE_k(G_s^{pub})$.

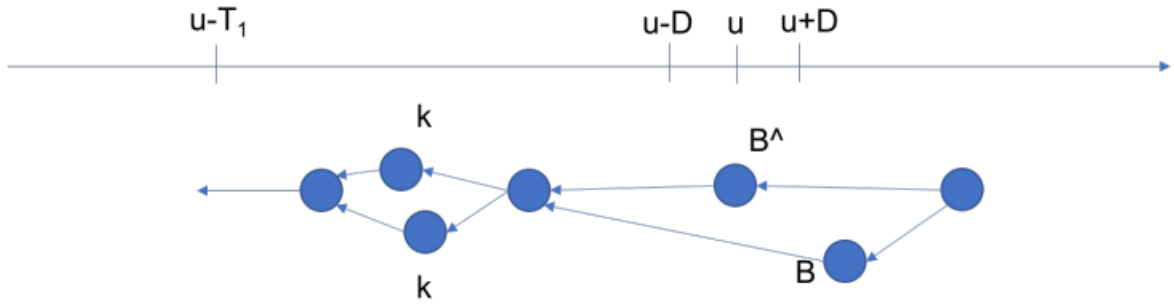

Рисунок 3.2 – Контрприклад до Claim 1 та Claim 3 (Part III)

Зауваження. Враховуючи визначення першої властивасті події $\mathcal{E}(t_0)$, дане твердження виконується до деякого часу u, а не для будь-якого $s \geq u$. Справа в тому, що Claim 1 вірне для деякого інтервалу у часі з діаметром D, але воно не доведене для загального випадку – до часу генерації наступного за $\hat{B}$ блоку. Контрприклад подано на рисунку 3.2. Припускаємо, що 2 блоки було згенеровано після $\hat{B}$, в той час B не бачить $\hat{B}$ під час своєї генерації, таким чином стає можливою ситуація, коли блок $\hat{B}$ синій і має синій блок у $anticone(\hat{B})$.

Твердження 3.3. *Перші 2 умови з означення події $\mathcal{E}(t_0)$ означають 3, тобто $score(Chn(\hat{B}))$ події завжди буде більше, ніж score будь якої гілки, що не проходить через $\hat{B}$. Це твердження є твердженням 3 в [9].*

Недолік 2

В останньому параграфі на сторінці 20 [9] йдеться, що поки $\hat{B}$ синій, тільки блоки зловмисника можуть впливати на score гілки згенерованої зловмисником. Тим не менш, блок злоумисника може посилатися на k-блоки створені в інтервалі $[u - T_1, u - D]$, уникаючи $\hat{B}$ (рисунок 3.2).

Недолік 3 Ми вважаємо що твердження 3.3 є хибним. На противагу цьому твердженню ми пропонуємо сценарій при якого зловмисник генерує

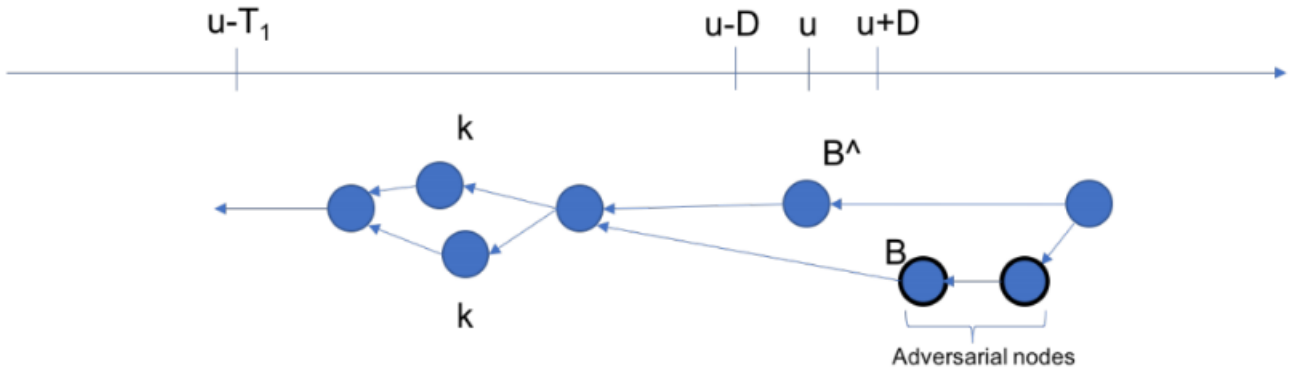

Рисунок 3.3 – Контрприклад до Claim 3

2 блоки після $u + D$ до створення чесного блоку. $score(Chn(\hat{B}))$ більше не є максимальним (рисунок 3.3).

Нами висунуто припущення, що насправді твердження мало б наступний вигляд: поки $\hat{B}$ – синій, жоден блок злоумисника, що було згенеровано до блоку $\hat{B}$ не буде сприяти гілці злоумисника (оскільки у них буде синій $(k + 1) - anticone$).

Недолік 4

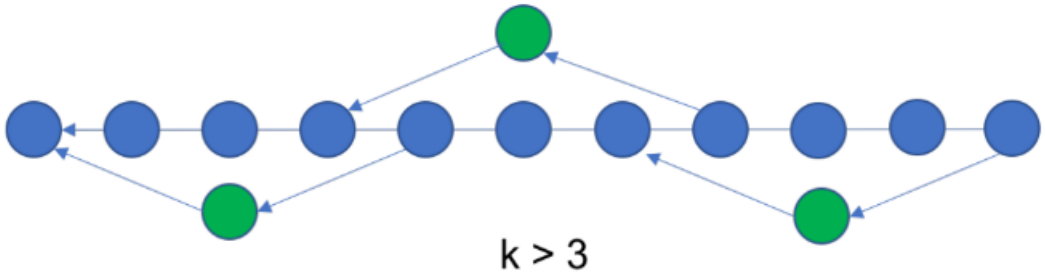

Рисунок 3.4 – Приклад з нечастими hourglass блоками

Сумнівним є твердження авторів про те, що **Hourglass блоки є частими**. Нехай маємо велике $k > 3$, на рисунку 3.4 зловмисник, генерує зелені блоки, сині блоки згенеровані чесними майнерами. Маючи α близьке до $\frac{1}{2}$ зловмисник може підтримувати зображений паттерн протягом тривалого проміжку часу, підтримуючи низьку частоту появи hourglass блоків.

Враховуючи вищесказане, не гарантовано, що блок $\hat{B}$ є синім. Тим не менш, явного аналізу ймовірності того, що блок $\hat{B}$ червоний, не дано – не зважаючи на те, що імовірність низька.

Недолік 5

Твердження 3.4. *Час появи події $\mathcal{E}(t_0)$ скінчений та обмежено зверху деякою константою, що не залежить від t_0. Дане твердження є твердженням 4 в [9].*

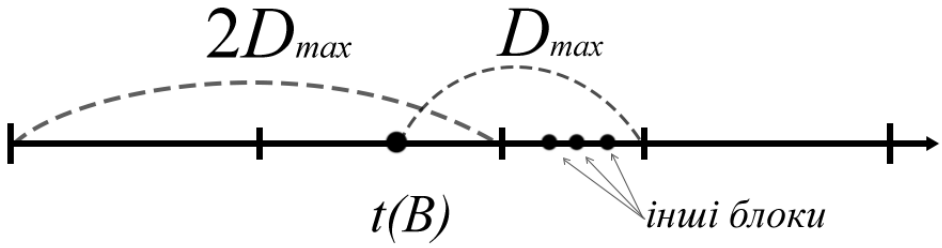

Рисунок 3.5 – Приклад з нечастими hourglass блоками

Нехай $time(B)$ – час генерації блоку B. У доведенні твердження подано йомовірність того, що в інтервалі $[time(B) - D_{max}, time(B) + D_{max}]$ не буде згенеровано жодного блоку окрім блоку B дорівнює

$$p = (1 - e^{-2D_{max}\lambda})^{-1} \cdot (1 - \sum_{j=2}^{\infty} e^{-2D_{max}\lambda} \cdot \frac{(e^{-2D_{max}\lambda})^j}{j!}).$$

В цьому виразі є друкарська помилка, оскільки генерація піддається розподілу Пуассону, тож на формула має вигляд:

$$p = (1 - e^{-2D_{max}\lambda})^{-1} \cdot (1 - \sum_{j=2}^{\infty} e^{-2D_{max}\lambda} \cdot \frac{(-2D_{max}\lambda)^j}{j!}). \qquad (3.9)$$

Але навіть з цим виправленням формула не дає ймовірності події $\mathcal{E}(t_0)$. Подія $\mathcal{E}(t_0)$ означає наступне:

1) блок B створено в момент часу $time(B)$;

2) в інтервалі $[time(B) - D_{max}, time(B) + D_{max}]$ не створено жодного іншого блоку.

За формулою 3.9 обчислюється умовна ймовірність того, що протягом інтервалу $[time(B) - D_{max}, time(B) + D_{max}]$ створено один блок *за умовою* того, що принаймні 1 блок створено в цьому інтервалі. Таким чином, блок може бути створено на інтервалі довжиною $2D_{max}$, що не означає, що протягом D_{max} секунд до $time(B)$ та D_{max} секунд після $time(B)$ не було інших блоків (рисунок 3.5).

3.3.2 Оцінки часу появи «пісочних годинників»

Нами виведено верхня та нижня оцінки часу до появи «пісочних годинників». Для доведення відповідних теорем нам знадобляться наступні позначення та означення. Нехай $\mathcal{A} = \{A_i\}_{i \geq 1}$ – множина подій; ξ_t – однорідний випадковий процес Пуассона, описуючий кількість подій з множини $\mathcal{A}$ на інтервалі довжини t (в розглянутих прикладах A_i – генерація блоку):

$$P(\xi_t = k) = \frac{(\lambda t)^k e^{-\lambda t}}{k!}. \tag{3.10}$$

Для довільної подій $A_i \in \mathcal{A}$ позначимо $t(A_i)$ – момент появи події A_i. Нехай $D > 0$ – деяка константа.

Визначення 3.2. Будемо казати, что сталася *D-ізольована подія* $A \in \mathcal{A}$, якщо на півінтервалах $[t(A) - D)$ та $(t(A) + D]$ не сталося жодної події з $\mathcal{A}$. В даному випадку $D-$ізольованими подіями є поява «пісочних годинників».

Позначимо випадкову величину τ, як мінімальне $T \geq 0$, що на інтервалі $[0, T]$ сталася хоча б одна $D-$ізольована подія.

Теорема 3.2. *Для будь-якого $T \geq 0$, D ділить T виконується*

наступне:

$$P(\tau \geq T) \geq (1 - e^{-\lambda D}(1 + \lambda D)).$$

Доведення.

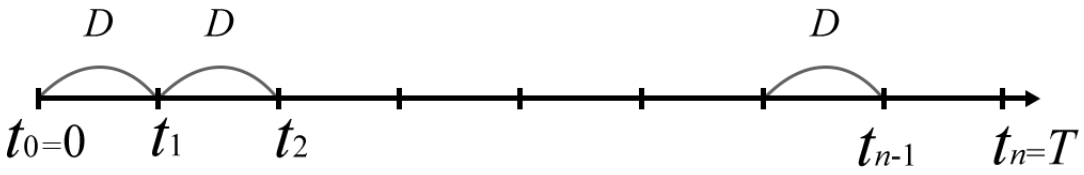

Рисунок 3.6 – Розбиття інтервалу $[0, T]$ на відрізки довжиною D

Для простоти припустимо, що $T = nD, n \in \mathbb{N}$, розіб'ємо відрізок $[0, T]$ на відрізки довжиною D т відмітимо відповідні точки: $t_0 = 0$, $t_i = t_{i-1} + D$, $i = \overline{1, n}$, де $n = \frac{T}{D}$ (рисунок 3.6). Позначимо $\hat{B}$ – подія, що до моменту T, сталася хоча б одна $D-$ізольована подія. Таким чином $\hat{B} = \{\tau \leq T\}$, $\neg\hat{B} = \{\tau > T\}$. Також позначимо $B_i, i = \overline{1, n}$, події, при якій на інтервалі $[t_{i-1}, t_i]$ сталося не менше двох подій з множини $\mathcal{A}$. Тепер доведемо наступне:

$$\cap_{i=1}^{n} B_i \subset \neg\hat{B} \tag{3.11}$$

Включення 3.11 будемо доводити від противного. Нехай сталася подія $\cap_{i=1}^{n} B_i$, але при цьому не сталася подія $\neg\hat{B}$, чи, що теж саме, сталася подія $\hat{B}$ (рисунок 3.7).

Нехай A – $D-$ізольована подія, $t(A) \in (t_{i-1}, t_i)$. Відмітимо, що $\forall i = \overline{0, n} : P(t(A) = t_i) = 0$. Таким чином на інтервалі $(t(A) - D, t(A))$ не сталося жодної події, відповідно на інтервалі $(t_{i-1}, t(A))$ – також, оскільки $(t_{i-1}, t(A)) \subset (t(A) - D, t(A))$ (рисунок 3.8).

Аналогічно на інтервалах $(t(A), t(A) + D)$ та $(t(A), t_i)$ також не

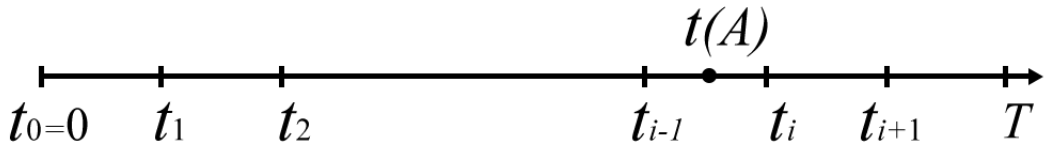

Рисунок 3.7 – Подія A на відрізку $[0, T]$

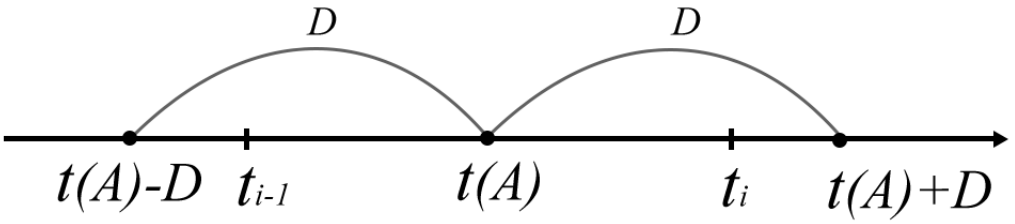

Рисунок 3.8 – Подія A на інтервалі (t_{i-1}, t_i)

сталося жодної події з $\mathcal{A}$.

Таким чином, на інтервалі (t_{i-1}, t_i) сталася рівно 1 поді, що суперечить припущенню про те, що сталася подія B_i. Включення 3.11 доведено.

Далі з 3.11 випливає, що

$$P(\neg\hat{B}) \geq P(\cap_{i=1}^{n} B_i). \tag{3.12}$$

Оскільки B_i – незалежні події, то

$$P(\neg\hat{B}) \geq \prod_{i=1}^{n} P(B_i). \tag{3.13}$$

Враховуючи означення подій $B_i, \overline{1,n}$ та формули 3.10 отримуємо:

$$P(B_i) = P(\xi_D \geq 2) = 1 - e^{-\lambda D}(1 + \lambda D). \tag{3.14}$$

Підставляючи 3.14 в 3.13 й використовуючи 3.12, отримуємо

$$P(\tau > T) = P(\neg\hat{B}) \geq (1 - e^{-\lambda D}(1 + \lambda D)),$$

що доводить теорему. □

Теорема 3.3. *Для довільного $\gamma > 0$ справедливою є оцінка*

$$P(\tau \geq T) \leq (1 - 2\gamma e^{-\lambda(2D+\gamma)})^{\frac{T}{2D+\gamma}}$$

Доведення.

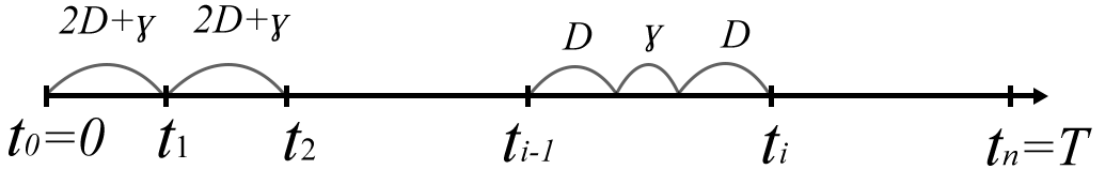

Рисунок 3.9 – Розбиття відрізка $[0, T]$ на відрізки довжиною $2D + \gamma$

Для довільного $\gamma > 0$ розіб'ємо відрізок $[0, T]$ на відрізки довжиною $2D + \gamma$ та відмітимо відповідні точки: $t_0 = 0$, $t_i = t_{i-1} + 2D + \gamma$, $i = \overline{1,n}$, для простоти будемо вважати, що $T = (2D + \gamma)n$ (рисунок 3.9).

Нехай B_i – подія, за якої на відрізку $[t_{i-1}, t_i]$ сталося хоча б одна D-ізольована подія; $B_i^{'}$ – подія, за якої на інтервалах $(t_{i-1}, t_{i-1} + D)$ та $(t_i - D, t_i)$ не сталося жодних подій з $\mathcal{A}$, а на інтервалі $(t_{i-1} + D, t_i - D)$ – сталася рівно одна подія з $\mathcal{A}$ (рисунок 3.10).

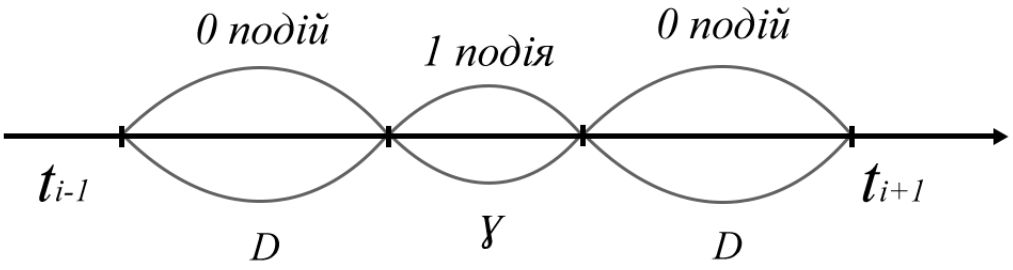

Рисунок 3.10 – Подія A на інтервалі $(t_{i-1} + D, t_i - D)$

Очевдно, що $B_i \subset B_i^{'}$, відповідно

$$P(B_i) \geq P(B_i^{'}) = e^{-\lambda D}(e^{-\lambda\gamma})e^{-\lambda D} = e^{-\lambda(2D+\gamma)}, i = \overline{1,n}. \tag{3.15}$$

Далі відмітимо, що $\neg\hat{B} \subset \prod_{i=1}^{n}(\neg B_i)$, до того ж події B_i – незалежні. Відповідно, враховуючи 3.15:

$$P(\neg\hat{B}) \leq \prod_{i=1}^{n}(\neg B_i) = \prod_{i=1}^{n}(1 - P(B_i)) \leq \prod_{i=1}^{n}(1 - P(B_i^{'})) =$$

$$= \prod_{i=1}^{n}(1 - 2\gamma e^{\lambda(2D+\gamma)}) = (1 - 2\gamma e^{\lambda(2D+\gamma)})^n = (1 - 2\gamma e^{\lambda(2D+\gamma)})^{\frac{T}{2D+\gamma}}.$$

Теорему доведено. □

3.3.3 Інші недоліки протоколу PHANTOM

Розгляне деякі інші, не настільки значущі проблеми специфікації протоколу:

1) Для кількісного аналізу, автори протоколу посилаються на протокол SPECTRE [8], що є помилкою, оскільки там аналіз також не вичерпний і детальний.

2) Враховуючи наш аналіз тверджень Claim 1 та Claim 3, виявляється, що у протоколу не пропонує надійного способу підтвердження блоків. Підтвердження блоків побудовано на основі «гарних» подій, які не обов'язково явно помічаються (наприклад, hourglass блоки будуть з'являтися за певних обставин, коли з'явилися k чесно згенерованих блоків в той час, коли не було згенеровано нечесних блоків).

3) Транзакції, що впорядковано в «деякому топологічному порядку». Якщо дану умову не буде чітко описано, це дасть змогу нечесному егоїстичні майнери переопубліковують транзакцію, що дає перевагу зловмиснику (рисунок **??**). Чесний майнер публікує блок B_h. Потім злоумисник публікує блок B_a, що містить конфліктуючі транзакції з B_h. Якщо «довшим» ланцюгам не будет надано приорітет у злоумисника буде розумна ймовірність успіху атаки.

4) Алгоритм 2.2 (Algrithm 1) на стічці 8 пастулює додавання блоків з антикону всупереч операціїї з рисунка 2.5. Що, скоріше за все, означає додавання блоку з минулого вузла, а не з його антикону. Дійсно, додавання з антикону може блокувати прогрес. Для цього припустимо, що $k+1$ блок було паралельно змайнено одразу після генезису. Припустимо, що будь-який з цих блоків з них не було обрано як наступний. Усі інші блоки теж стануть блакитником. Будь-який з наступних блоків (з глибиною 2) не зможе стати синім, оскільки в нього буде синій антикон потужності $(k+1)$.

3.4 Недоліки протоколів GHOST і Tangle

Після попереднього аналізу протоколів GHOST і Tangle нами були виявлені деякі недоліки у доведеннях тверджень.

3.4.1 Недолік протоколу GHOST

Нехай A – зловмисник, λ – швидкість генерації блоків усією мережею, λ_{rep} – швидкість генерації блоків, що спостерігається учасниками мережі, λ_a – швидкість генерації блоків зловмисником, λ_h – швидкість генерації блоків чесними майнерами, $\beta = \beta(\lambda)$ – швидкість росту основного ланцюжка, q – частка обчислювальної потужності зловмисника.

Твердження 3.5. *Якщо* $\beta(\lambda_{rep}) > \frac{q}{1-q}\lambda$, *то мережа захищена від* q*-атаки. Більш того, зловмисник буде найбільш ефективним за умови, якщо він не буде генерувати блоки до початку атаки. Дане твердження є твердженням 3.1 в специфікації протоколу GHOST [10].*

Зауваження. Розглянемо доведення, що пропонується авторами протоколу. Нехай f – доля A, що задіюється до атаки. Отримуємо наступні співвідношення:

$$\lambda_{rep} = \lambda_h + f\lambda_a \Rightarrow \lambda_{rep} = \lambda_h + f\lambda_a = \lambda - f\frac{q}{1-q}\lambda_h \Rightarrow$$

$$\Rightarrow \lambda_h = \frac{\lambda_{rep}}{1 + f\frac{q}{1-q}}$$

Далі отримане співвідношення підставляється в $\beta(\lambda_h)$:

$$\beta(\lambda_h) = \beta(\frac{\lambda_{rep}}{1 + f\frac{q}{1-q}}).$$

Автори стверджуть, що $\beta(\frac{\lambda_{rep}}{1+f\frac{q}{1-q}}) \geq \beta(\lambda_{rep})$, що є безпідставним твердженням.

3.5 Недоліки протоколу Tangle

Недолік 1

У цій секції 3.1 роботи [7] автори протоколу намагаються оцінити швидкість росту кумулятивної ваги транзакції. Розглянемо викладки авторів протоколу стосовно режиму високого навантаження. Автори припускають, що кумулятивна вага транзакції, що була додана давно і вже підтверджена, росте зі швидкістю λ, λ – швидкості росту Tangle, для нових транзакцій вони пастулюють, що швидкість росту кумулятивної ваги спочатку меньшою за λ, а потім зростає до λ.

Зауваження. Твердження стосовно поступового росту швидкості збільшення комулятивної ваги транзакції не доведено строго. Також немає строгого доведення твердження про комулятивну вагу вже підтвердженної транзакції.

Недолік 2

Далі автори намагаються оцінити ймовірність того, що нова транзакція згенерована в момент t дає хоча б одне підтвердження одній з «наших» tips (ЧТО ТАКОЕ НАШИ TIPS). Для цього вони вводять наступні позначення: нехай $H(t)$ – середня очікувана кумулятивна вага транзакції в момент часу t, $K(t)$ – середня кількість tips, котрі підтверджують транзакцію в момент часу t. Для спрощення, автори роблять припущення, що швидкість додавання нових транзакції є (майже) константною і дорівнює L_0. Для спрощення автори позначають величину $h := h(L_0, N)$.

Зауваження. У позначенні величини h не відомо, яку величину позначено змінною N.

Недоліки 3, 4

Отже, шукана ймовірність на думку авторів складає:

$$1 - (1 - \frac{K(t-h)}{L_0})^2 = \frac{K(t-h)}{L_0}(2 - \frac{K(t-h)}{L_0}).$$

Зауваження. Дана формула була б вірною, якщо б $K(t-h)$ було б рівно кількості транзакцій, що були видимі в момент t (может тут $t-h$) и залишились непідтвердженими в момент t.

Зауваження. Далі автори намагаються обрахувати значення $K(t)$. Математичні викладки також не є вичерпними. Наприклад у нас виникли питання, при отриманні рівняння (5) з рівняння (4) в роботі:

$$(p_1 - p_2)\lambda \approx \lambda \frac{K(t-h)}{L_0} = \frac{K(t-h)}{2h}$$

Але не зрозуміло, чому дане визначення рівне $\frac{dK(T)}{dt}$, адже не було наведено жодних викладок стосовно цього граничного переходу.

Аналіз протоколів Tagle та GHOST є дещо поверхневим, проте навіть знайдені недоліки свідчать про наявність помилок в обгрунтуваннях стійкості протоколів. Подальший аналіз може стати темою наступних робіт.

3.6 Висновки до розділу 3

У даному розділі детально розглянуто 2 основних види атак на протоколи криптовалют. Наведено детальний аналіз недоліків протоколів Blockchain-free та PHANTOM. Показано, що протоколи мають суттєві недоліки в обгрунтуваннях своєї стійкості. Наведено атаку на протокол Blockchain-free з імовірністю успіху близькою до 1. Уточнено верхню та нижню границю часу прийняття блоків мережею PHANTOM. Зроблено первинний аналіз протоколів GHOST і Tangle, в яких також були виявлені недоліки при доведеннях тверджень.

ВИСНОВКИ

В даній роботі нами проведено аналіз недоліків технології блокчейн на шляху до її масштабування. Зроблено висновок, що підвищення пропускної здатності шляхом збільшення швидкості генерації нових блоків чи їх укрупнення призводить до зниження стійкості до атак подвійної витрати. Інші способи підвищення пропускної здатності (компресія блоків, застосуванням таблиць пошуку Блума та ін.) не дають необхідного приросту продуктивності блокчейну. Таким чином гостро стоїть потреба у нових підходах до побудови протоколів криптовалют.

Нами наведено опис технології графчейн, що є узагальненням технології блокчейну. Зазначено, які переваги й недоліки має така конструкція. Описано значну кількість протоколів консенсусу, які пропонуються для графчейну.

Нами показано, що всі розглянуті протоколи мають недоліки в обгрунтуваннях стійкості, тож жоден з них на даний час не має доведеної стійкості до класичних атак, наприклад, до атаки подвійної витрати. Побудовано ефективну атаку на протокол Blockchain-free [5], представлено аналітичні оцінки її часу підготовки в залежності від частки обчислювальних ресурсів, що зосереджено у злоумисиника. Уточнено оцінки часу прийняття блоків у протоколі PHANTOM [9].

ПЕРЕЛІК ПОСИЛАНЬ

1. Dwork, C., Naor, M. Pricing via processing or combatting junk mail. CRYPTO (1993)

2. Jakobsson, Markus; Juels, Ari (1999). "Proofs of Work and Bread Pudding Protocols". Secure Information Networks: Communications and Multimedia Security. Kluwer Academic Publishers: 258–272.

3. Nakamoto, Satoshi (31 October 2008). "Bitcoin: A Peer-to-Peer Electronic Cash System"(PDF). bitcoin.org. Archived (PDF) from the original on 20 March 2014. Retrieved 28 April 2014.

4. Nakamoto; et al. (1 April 2016). "Bitcoin source code - amount constraints". Archived from the original on 1 July 2018.

5. Blockchain-Free Cryptocurrencies: A Framework for Truly Decentralised Fast Transactions: https://eprint.iacr.org/2016/871.pdf

6. Kyle Butt, Derek Sorensen, and Michael Stay: Casanova 2019

7. The Tangle Serguei Popov https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvs April 30, 2018. Version 1.4.3

8. SPECTRE: Serialization of Proof-of-work Events: Confirming Transactions via Recursive Elections Yonatan Sompolinsky, Yoad Lewenberg, and Aviv Zohar https://eprint.iacr.org/2016/1159.pdf 2016

9. PHANTOM and GHOSTDAG A Scalable Generalization of Nakamoto Consensus February 2, 2020

10. Secure High-Rate Transaction Processing in Bitcoin (full version) 2013

11. https://usa.visa.com/run-your-business/small-business-tools/retail.html

12. L. Kovalchuk, D. Kaidalov, O. Shevtsov, A. Nastenko, M. Rodinko and R. Oliynykov, "Analysis of splitting attacks on Bitcoin and GHOST consensus protocols,"2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and

Applications (IDAACS), Bucharest, 2017, pp. 978-982.

13. Aggelos Kiayias ,Alexander Russell ,Bernardo David and Roman Oliynykov Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol. https://eprint.iacr.org/2016/889.pdf

14. Jehyuk Jang and Heung-No Lee, Profitable Double-Spending Attacks, https://arxiv.org/pdf/1903.01711.pdf

15. Pinzón, Carlos Rocha, Camilo, Double-spend Attack Models with Time Advantange for Bitcoin, 2016, Theoretical Computer Science. 329. 79-103. 10.1016/j.entcs.2016.12.006.

16. A. Pinar Ozisik, George Bissias, Brian N. Levine, Estimation of Miner Hash Rates and Consensus on Blockchains, 2017, https://arxiv.org/abs/1707.00082